

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

THOMÁS AUGUSTO HENNIG

**VALIDAÇÃO DO MÉTODO LPC PARA VOCODER  
MUSICAL E ANÁLISE DOS EFEITOS DE QUANTIZAÇÃO  
COM PONTO FIXO**

Porto Alegre  
2017

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
ESCOLA DE ENGENHARIA  
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

THOMÁS AUGUSTO HENNIG

**VALIDAÇÃO DO MÉTODO LPC PARA VOCODER  
MUSICAL E ANÁLISE DOS EFEITOS DE QUANTIZAÇÃO  
COM PONTO FIXO**

Projeto de Diplomação apresentado ao Departamento de Engenharia Elétrica da Escola de Engenharia da Universidade Federal do Rio Grande do Sul, como requisito parcial para Graduação em Engenharia Elétrica.

Orientador: Prof. Dr. Adalberto Schuck Júnior

Porto Alegre

2017

THOMÁS AUGUSTO HENNIG

**VALIDAÇÃO DO MÉTODO LPC PARA VOCODER  
MUSICAL E ANÁLISE DOS EFEITOS DE QUANTIZAÇÃO  
COM PONTO FIXO**

Este trabalho de conclusão foi analisado e julgado adequado para a obtenção do título de Bacharel em Engenharia Elétrica e aprovado em sua forma final pelo Orientador e pela Banca Examinadora designada pelo Departamento de Engenharia Elétrica da Universidade Federal do Rio Grande do Sul.

---

Prof. Dr. Adalberto Schuck Júnior

---

Prof<sup>a</sup> Dr<sup>a</sup> Leia Bernardi Bagesteiro

Aprovado em: \_\_\_\_/\_\_\_\_/\_\_\_\_

BANCA EXAMINADORA

Prof. Dr. Adalberto Schuck Jr.– UFRGS

Prof. Dr. Ronaldo Husemann – UFRGS

Prof. Dr. Tiago Roberto Balen – UFRGS

---

---

---

*A todos que dedicaram a vida à música e a trataram com o respeito que ela merece.*

## **AGRADECIMENTOS**

Agradeço à minha família que, além de fornecer o suporte e estrutura necessários para minha formação, sempre reforçou a importância da busca pelo meu desenvolvimento cultural e pessoal. Agradeço também ao meu orientador, Adalberto Schuck Júnior, pela paciência e boa vontade que apresentou ao longo da realização do trabalho. Por fim, agradeço a todos cuja presença tornou mais feliz minha trajetória até aqui.

## RESUMO

A modelagem do sistema fonador humano como um sistema fonte-filtro permitiu, com a reprodução eletrônica deste sistema, a criação sintética de fala. Apesar de muito disseminada nos sistemas de telecomunicação, a sintetização de voz através de predição linear não apresenta a mesma popularidade na construção de VOCODERs de aplicação musical. Visando a futura construção de um VOCODER musical, este trabalho busca validar o método LPC (*Linear Predictive Coding*) para tal aplicação e verificar os efeitos da quantização dos sinais de entrada no processo. Através de simulações no *MATLAB*, é realizada a análise LPC do sinal de fala e aplicado um sinal musical como excitação do filtro resultante. Através da avaliação quantitativa, com gráficos dos resultados, e qualitativa, ouvindo o sinal sintetizado, é verificada a viabilidade do método. Para verificação dos efeitos da quantização, compara-se os resultados do procedimento quantizado utilizando ponto fixo, com tamanho de palavra de 8, 16 e 32 bits, com os resultados obtidos no modo de alta precisão, utilizando variáveis do tipo *double*. Os resultados obtidos validam a aplicação do método LPC para aplicação musical, tendo como saída do sistema o sinal de fala “cantado” no tom do sinal de excitação. Do ponto de vista de quantização, nos testes realizados, somente as quantizações com 16 e 32 bits se mostraram viáveis, com erro de quantização RMS de 9,17% e 8,56%, respectivamente.

**Palavras-chave:** VOCODER; Análise LPC; Síntese LPC; Predição Linear.

## ABSTRACT

The modeling of the human vocal system as a source-filter system allowed, with its electronic implementation, the synthetic creation of speech. Although speech synthesis by linear prediction is often used in telecommunication systems, this method is not as popular in musical VOCODERs construction. Aiming the construction of a musical VOCODER, the objective of this work is to validate the LPC (Linear Predictive Coding) method for this application and to evaluate the effects of the quantization of the input signals. Through *MATLAB* simulations, the LPC analysis of the input speech signal is performed and a musical signal is applied as the system's excitation. The method's viability is verified through quantitative evaluating, plotting the results, and qualitative evaluation, listening to the synthesized signal. The quantization effects are evaluated by comparing the results with fixed-point quantization, using 8, 16, and 32 bits word lengths, with the results using double precision variables. The results obtained in this work confirm the LPC method viability for musical application, since the obtained system's output is the speech signal "singed" in the excitation signal pitch. The quantization proved to be feasible only with 16-bit and 32-bit word lengths, presenting, respectively, a 9,17% and 8,16% RMS quantization error.

**Keywords:** VOCODER; Linear Prediction; LPC Analysis; LPC Synthesis.

## LISTA DE ILUSTRAÇÕES

Figura 1 – VOCODER do projeto SIGSALY, fotografado em 1944.....	15
Figura 2 – EMS Synthi 100 VOCODER, criado em 1971.....	16
Figura 3 – VOCODER de 10 bandas, publicado pela revista Elektor em 1980.....	16
Figura 4 – Diagrama do banco de filtros do VOCODER da Elektor. ....	17
Figura 5 – (a) Roland VP-330. (b) Roland VP-03 .....	18
Figura 6 - Anatomia do sistema fonador humano. ....	20
Figura 7 – Sinal de passagem de ar pela glote para diferentes pressões subglotais. ....	21
Figura 8 – Sinal de fala no domínio do tempo e de frequência. ....	22
Figura 9 – Espectrograma das palavras <i>heed, hid, head, had, hod, hawed, hood, who'd</i> pronunciadas por um americano do sexo masculino. ....	23
Figura 10 – Três primeiras formantes das vogais pronunciadas na Figura 9. ....	23
Figura 11 – Modelo fonte-filtro.....	24
Figura 12 – Sistema de análise e síntese LPC. ....	26
Figura 13 – Diagrama proposto de VOCODER musical. ....	27
Figura 14 – Círculo de complemento de dois para uma número de 3 bits (a) inteiro e (b) fracionário.....	34
Figura 15 – Exemplo de quantização de sinal senoidal.....	35
Figura 16 – Distribuição de probabilidade do erro de quantização. ....	36
Figura 17 – Possíveis posições no plano z dos polos para o filtro da equação (55) quantizado com 4 bits.....	39
Figura 18 – Implementação direta de par de polos complexos conjugados. ....	40
Figura 19 – Posições possíveis dos polos do sistema mostrado na Figura 18 quantizado com (a) 4 bits e (b) 7 bits.....	40
Figura 20 – Distribuição da frequência fundamental da voz masculina e feminina.....	42
Figura 21 – Sinal de excitação do sistema.....	43
Figura 22 – Espectro de frequência do sinal de excitação do sistema.....	44
Figura 23 – Janela de Hanning aplicada (domínio do tempo).....	45
Figura 24 – Janela de Hanning aplicada (domínio da frequência). ....	45
Figura 25 – Sobreposição da janela de Hanning aplicada. ....	46
Figura 26 – Frame de fala de 32ms. ....	51
Figura 27 – Densidade espectral do <i>frame</i> . ....	51

Figura 28 – Resposta em frequência do filtro LPC. ....	52
Figura 29 – Polos do filtro LPC no plano-z.....	52
Figura 30 – Sobreposição do formato da resposta do filtro LPC à densidade espectral do <i>frame</i> . ....	53
Figura 31 – Trem de impulsos de 111 Hz aplicado como excitação para síntese LPC.....	53
Figura 32 – Comparação entre <i>frame</i> original de fala e <i>frame</i> sintetizado.....	54
Figura 33 – Comparação entre densidade espectral do <i>frame</i> original e do <i>frame</i> sintetizado.	54
Figura 34 – Densidade espectral do <i>frame</i> de excitação. ....	55
Figura 35 - Densidade espectral do <i>frame</i> sintetizado.....	55
Figura 36 – Polos alocados no primeiro quadrante do plano-z. ....	56
Figura 37 - Polos alocados no segundo quadrante do plano-z. ....	57
Figura 38 - Polos alocados no terceiro quadrante do plano-z. ....	57
Figura 39 - Polos alocados no quarto quadrante do plano-z. ....	58
Figura 40 – Frames de fala e excitação quantizados com 8 bits e sua densidade espectral. ....	59
Figura 41 – Filtro LPC frame quantizado com 8 bits. ....	59
Figura 42 – Sobreposição de polos quantizados com 8 bits e não quantizados no plano-z ....	60
Figura 43 – Comparação entre <i>frame</i> sintetizado com 8 bits e sintetizado sem quantização...60	
Figura 44 – Densidade espectral do erro de quantização com 8 bits.....	61
Figura 45 – <i>Frames</i> de fala e excitação quantizados com 16 bits e sua densidade espectral...61	
Figura 46 – Filtro LPC quantizado com 16 bits. ....	62
Figura 47 – Sobreposição de polos quantizados com 16 bits e não quantizados no plano-z. ..62	
Figura 48 – Comparação entre frame sintetizado com 16 bits e sintetizado sem quantização. 63	
Figura 49 – Densidade espectral do erro de quantização de 16 bits.....	63
Figura 50 – Frame de fala quantizado com 32 bits e sua densidade espectral. ....	64
Figura 51 – Filtro LPC quantizado com 32bits. ....	65
Figura 52 – Sobreposição de polos quantizados com 32 bits e não quantizados no plano-z. ..65	
Figura 53 – Comparação entre frame sintetizado com 32 bits e sintetizado sem quantização. 66	
Figura 54 – Densidade espectral do erro da quantização de 32 bits.....	66

## LISTA DE TABELAS

Tabela 1 – Coeficientes LPC do <i>frame</i> analisado.....	52
Tabela 2 - Coeficientes LPC do <i>frame</i> quantizado com 8 bits. ....	59
Tabela 3 - Coeficientes LPC do <i>frame</i> quantizado com 16 bits. ....	62
Tabela 4 - Coeficientes LPC do <i>frame</i> quantizado com 32 bits. ....	64

## LISTA DE SIGLAS

AR	Auto Regressivo
IIR	<i>Infinite Impulse Response</i>
LSB	<i>Least Significant Bit</i>
LPC	<i>Linear Predictive Coding</i>
MA	<i>Moving Average</i>
SNR	<i>Signal-to-Noise Ratio</i>
MSB	<i>Most Significant Bit</i>
VOCODER	<i>Voice Encoder</i>

## LISTA DE SÍMBOLOS

$\Sigma$	Somatório
$R(x)$	Autocorrelação
$\varepsilon$	Valor esperado
$\int$	Integral
$\varphi$	Covariância
$\alpha_n$	Enésimo coeficiente de predição
$\Delta$	Passo de quantização
$H(z)$	Resposta ao impulso
$\sigma$	Desvio padrão

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>15</b>
<b>2 FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>20</b>
2.1 SINTETIZAÇÃO DA FALA .....	20
2.1.1 O sistema fonador humano .....	20
2.1.2 Sintetização da fala humana - VOCODER.....	24
2.1.3 Aplicação musical.....	26
2.2 O MÉTODO LPC – CODIFICAÇÃO POR PREDIÇÃO LINEAR .....	27
2.2.1 O conceito de predição linear .....	27
2.2.2 O modelo <i>all-pole</i> de predição linear .....	28
2.2.3 Estimação no domínio do tempo .....	28
2.2.4 Estimação no domínio da frequência – estimação espectral .....	31
2.2.5 Determinação da ordem de predição .....	32
2.2.6 Obtenção dos parâmetros pelo Algoritmo de Levinson-Durbin.....	32
2.3 EFEITOS DA PRECISÃO NUMÉRICA LIMITADA EM SISTEMAS DIGITAIS .....	33
2.3.1 Representação de números em ponto fixo.....	33
2.3.2 Quantização .....	35
2.3.3 Quantização de coeficientes de filtros digitais .....	37
<b>3 MATERIAIS E MÉTODOS .....</b>	<b>42</b>
3.1 MATERIAIS .....	42
3.1.1 Sinal de fala e frequência de amostragem .....	42
3.1.2 Sinal de excitação musical do sistema.....	43
3.1.3 Janelamento e sobreposição.....	44
3.1.4 Algoritmo LPC .....	46
3.1.5 Síntese LPC .....	46
3.1.6 Quantização dos sinais de entrada .....	47
3.1.7 Modificação do algoritmo de análise LPC para funcionamento com ponto-fixa. ....	47
3.2 MÉTODOS .....	48

3.2.1 Prova de conceito de LPC para aplicação musical .....	48
3.2.1.1 Análise de frame de fala isolado.....	48
3.2.1.2 <i>Benchmark</i> do filtro LPC obtido .....	48
3.2.1.3 Síntese de <i>frame</i> único usando excitação musical.....	49
3.2.1.4 Análise e síntese musical com sinal de fala inteiro. ....	49
3.2.2 Análise dos efeitos de quantização .....	49
<b>4 RESULTADOS .....</b>	<b>51</b>
4.1 VALIDAÇÃO DO MÉTODO DE LPC PARA APLICAÇÃO MUSICAL .....	51
4.1.1 Análise LPC em <i>frame</i> de fala isolado. ....	51
4.1.2 <i>Benchmark</i> do filtro LPC.....	53
4.1.3 Síntese LPC com sinal de excitação musical em <i>frame</i> isolado.....	55
4.1.4 Análise e síntese musical com sinal de fala inteiro. ....	56
4.2 ANÁLISE DOS EFEITOS DE QUANTIZAÇÃO NA ANÁLISE LPC .....	58
4.2.1 Quantização de 8 bits.....	58
4.2.2 Quantização de 16 bits.....	61
4.2.3 Quantização de 32 bits.....	64
<b>5 CONCLUSÃO.....</b>	<b>68</b>
<b>6 REFERÊNCIAS BIBLIOGRÁFICAS.....</b>	<b>70</b>
<b>ANEXO A – ALGORITMO PARA EXPERIMENTO COM FRAME ÚNICO.....</b>	<b>72</b>
<b>ANEXO B – ALGORITMO PARA EXPERIMENTO COM SINAL INTEIRO.....</b>	<b>74</b>
<b>ANEXO C – ALGORITMO PARA EXPERIMENTO COM QUANTIZAÇÃO .....</b>	<b>76</b>

## 1 INTRODUÇÃO

Inicialmente criado para fins militares, o VOCODER (abreviação de *voice encoder*) representou uma nova maneira de transmissão de fala. Tendo sua criação creditada a Homer Dudley, da Bell Labs, em 1928, o VOCODER permitiu a recriação da voz humana de forma totalmente artificial, através de um sistema fonte-filtro (TOMPKINS, 2011). Tal possibilidade significou um ganho na eficiência de transmissão, uma vez que passou a ser necessário somente o envio de parâmetros para sintetização da fala e não mais o sinal de voz modulado.

Figura 1 – VOCODER do projeto SIGSALY, fotografado em 1944.



Fonte: Tompkins (2011)

Entretanto, se afastando dos objetivos que encorajaram sua criação, o VOCODER representou também uma novidade no cenário musical. Ao usar um instrumento musical como fonte de excitação na sintetização da voz, descobriu-se uma forma de fazer este instrumento “falar”. Em 1971, na estreia do filme *Laranja Mecânica* nos cinemas, o grande público teve seu primeiro contato com o VOCODER musical, ao ouvir a nona sinfonia de Beethoven “cantada” através de um VOCODER criado por Bob Moog e Wendy Carlos (TOMPKINS, 2011). Mais tarde, nesta década, os VOCODERs musicais seriam usados por diversos artistas, como os nomes máximos da música eletrônica Giorgio Moroder e Kraftwerk.

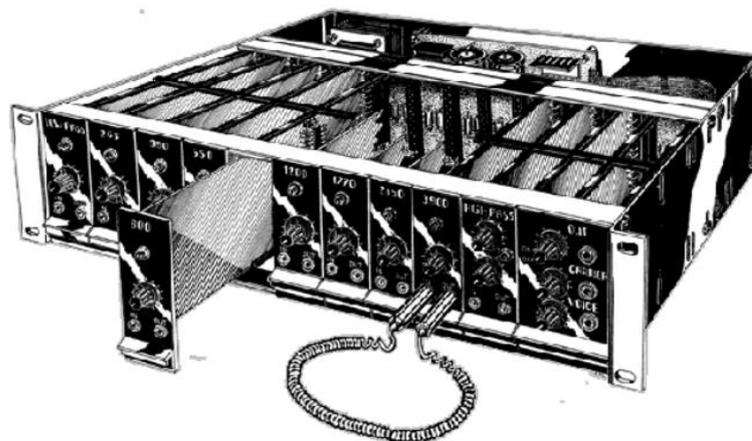
Figura 2 – EMS Synthi 100 VOCODER, criado em 1971.



Fonte: Tompkins (2011).

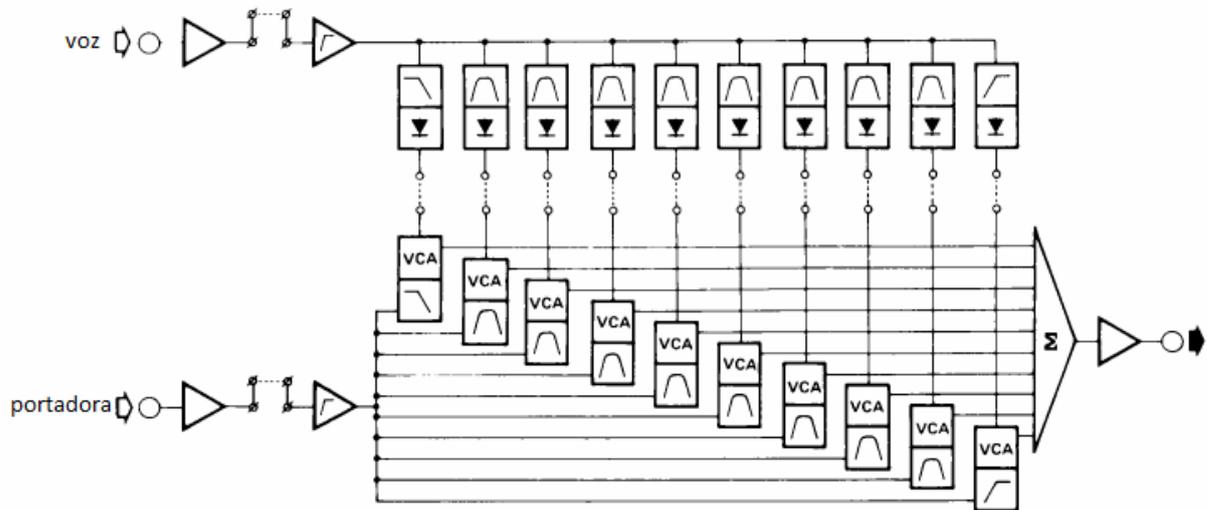
A sintetização da fala se dava, nessa época, de forma completamente analógica, o que resultava em aparelhos grandes com estruturas rudimentares. Uma das maneiras de realizar a análise e síntese de voz de forma analógica se dava através de grandes bancos de filtros. Esta era a maneira como funcionava, por exemplo, o VOCODER de 10 bandas proposto pela revista *Elektor* em 1980, mostrado na Figura 3 e na Figura 4.

Figura 3 – VOCODER de 10 bandas, publicado pela revista *Elektor* em 1980.



Fonte: *Elektor Magazine* (1980).

Figura 4 – Diagrama do banco de filtros do VOCODER da Elektor.



Fonte: Modificado de Elektor Magazine (1980).

O esquema mostrado na Figura 4, mostra dois bancos de filtros. O primeiro banco, responsável pela análise, divide a fala do usuário em faixas de frequência, que, de acordo com a característica da fala, recebem pesos representados pela tensão na saída de detectores de envelope. O segundo banco de filtros, por onde entra o sinal do instrumento, é idêntico ao primeiro e é responsável pela síntese do VOCODER. Nele, cada faixa de frequência é ligada em um VCA (*Voltage Controlled Amplitude*), um amplificador controlado por tensão, que usa como referência a tensão do detector de envelope do filtro correspondente no banco de análise. Dessa forma, o peso de cada faixa de frequência da voz é refletido no sinal do instrumento, dando a este instrumento características da fala (VISSER, 1980).

Com o advento das técnicas de processamento digital de sinais, os métodos de codificação de voz se alteraram. Se antes eram utilizados aparelhos caros, grandes e com estruturas rudimentares, o tratamento digital de sinais permitiu uma construção mais barata e métodos mais eficientes de codificação, substituindo, por exemplo, o banco de filtros para síntese por métodos de filtragem adaptativa. Dentre estas novas técnicas, está a codificação por LPC (*Linear Predictive Coding*), bastante utilizada para sintetização de fala humana.

Apesar de bastante utilizado para sintetização literal da fala, na pesquisa realizada não foi encontrado nenhum caso de aplicação do método LPC para VOCODERs musicais. Um dos fatores que certamente colabora para isso é a parcialidade, pendendo para sistemas analógicos, que existe no mercado de equipamentos musicais. Mesmo plataformas puramente

digitais acabam voltando o foco para a emulação de sistemas analógicos e, com VOCODERs, não é diferente. Os VOCODERs digitais mais populares são, na verdade, plataformas que, através de processamento digital de sinais, realizam a modelagem de sistemas analógicos. É o caso, por exemplo, do VP-03, da Roland, que propõe a replicação, com modelagem digital, do VP-330, VOCODER analógico de 10 bandas, da mesma empresa, lançado em 1979.

Figura 5 – (a) Roland VP-330. (b) Roland VP-03 .



Fonte: Site oficial da Roland (acessado em 2017).

Se por um lado a utilização de sistemas digitais representa uma alternativa mais prática e barata para codificação de voz, por outro, existe a necessidade de quantização de sinais analógicos. A quantização é um processo não linear e não reversível (PROAKIS; MANOLAKIS, 2007). Este processo introduz erro no sinal e causa, inevitavelmente, a perda de informação. Existe então um *trade off* entre qualidade e praticidade, o que faz com que uma análise dos efeitos da quantização seja ponto crucial na decisão do tipo de abordagem para implementação do sistema.

Assim, motivado pelo legado tecnológico e criativo do VOCODER e, principalmente, pela possibilidade de mesclar as áreas de música e engenharia, este trabalho tem como objetivos:

- Realizar uma prova de conceito e validar o método de codificação de voz por LPC para VOCODERs musicais.
- Analisar os efeitos da quantização no procedimento, avaliando a viabilidade de aplicação do método com diferentes resoluções na quantização.

O presente documento mostra, no capítulo dois, a fundamentação teórica por trás da sintetização de fala, realiza uma abordagem teórico-matemática do método LPC e discute

os efeitos da precisão numérica limitada em sistemas digitais. No terceiro capítulo são descritos os experimentos realizados para cumprimento dos objetivos listados acima e, no quarto capítulo, são mostrados os resultados obtidos. Por fim, no capítulo cinco, são discutidos os resultados e feitas as considerações finais.

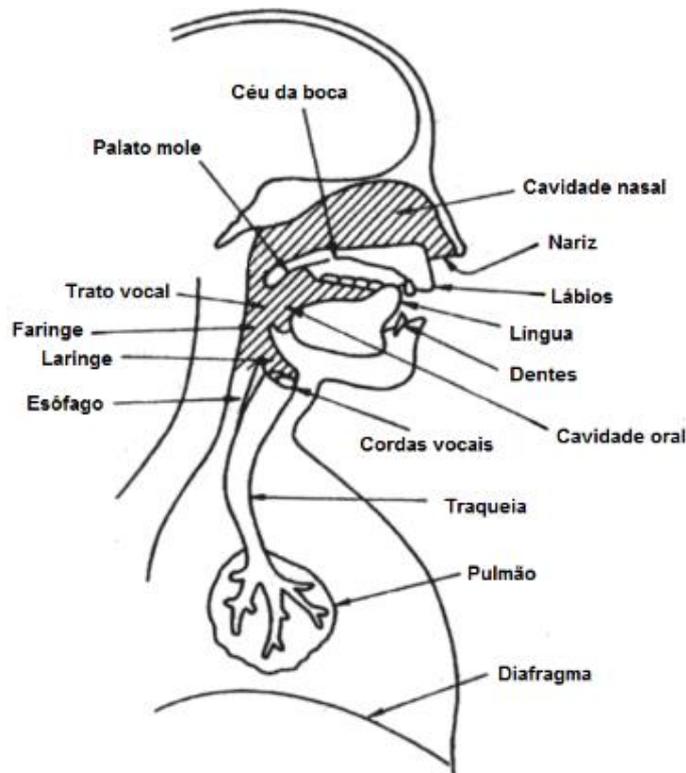
## 2 FUNDAMENTAÇÃO TEÓRICA

### 2.1 SINTETIZAÇÃO DA FALA

#### 2.1.1 O sistema fonador humano

A sintetização da fala humana tem como cerne a modelagem e reprodução artificial do aparelho fonador humano. Dessa forma, é conveniente uma breve explicação de como funciona este sistema. A Figura 6 mostra a anatomia do sistema fonador, indicando seus componentes.

Figura 6 - Anatomia do sistema fonador humano.

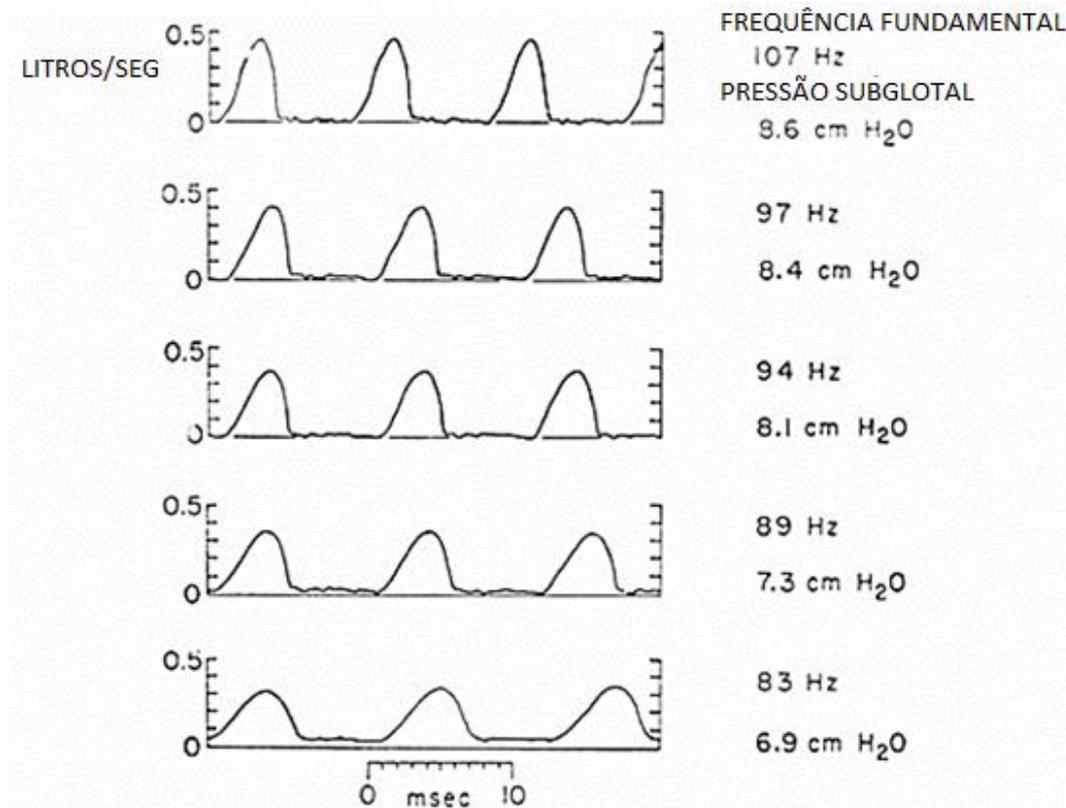


Fonte: Adaptado de Furui (2000).

Quando os músculos abdominais forçam o diafragma, o ar é empurrado para fora dos pulmões, criando um fluxo pela traqueia, glote e laringe (FURUI, 2000). A glote é o espaço entre as cordas vocais que, quando há intenção de produção de fala, se torna mais estreito. A passagem de ar é então periodicamente interrompida pela abertura e fechamento da glote, originando um sinal de pressão sonora com característica *quasi periódica*, como o mostrado

na Figura 7. A componente periódica da excitação da glote é caracterizada pela sua frequência fundamental, definindo o tom da voz (DUTOIT; MARQUES, 2010).

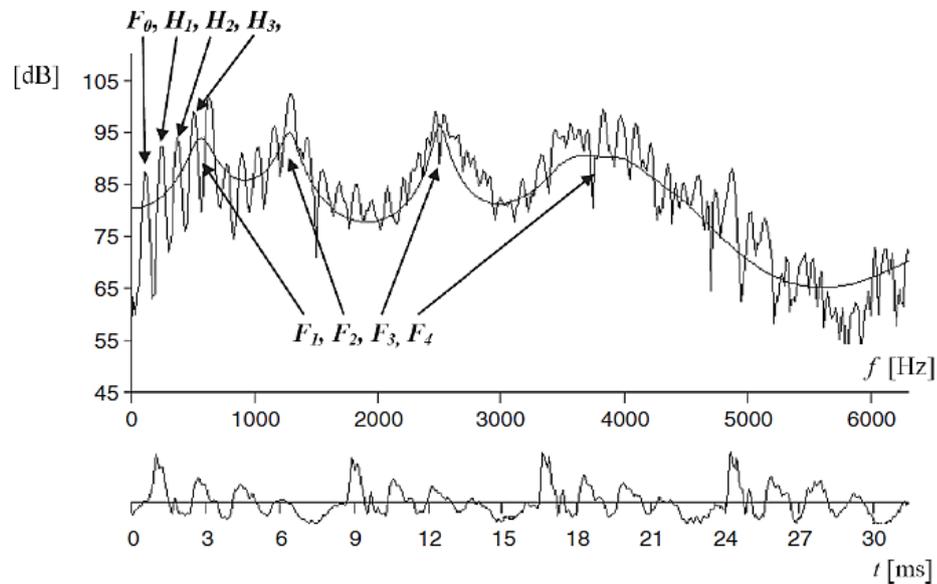
Figura 7 – Sinal de passagem de ar pela glote para diferentes pressões subglotais.



Fonte: Rothenberg (1971).

O sinal de pressão sonora causado pela passagem de ar pela glote passa então pelo trato vocal, que consiste de todo o caminho restante até as cavidades nasal e oral, indicado pela área hachurada na Figura 6. O formato do trato vocal define a ressonância do sinal gerado pelas cordas vocais. O movimento da boca, da língua e dos músculos altera o formato do trato vocal, alterando suas frequências de ressonância e permitindo a geração de diferentes sons. A Figura 8 mostra um frame de 30 milissegundos de um sinal de fala e seu espectro de frequência.

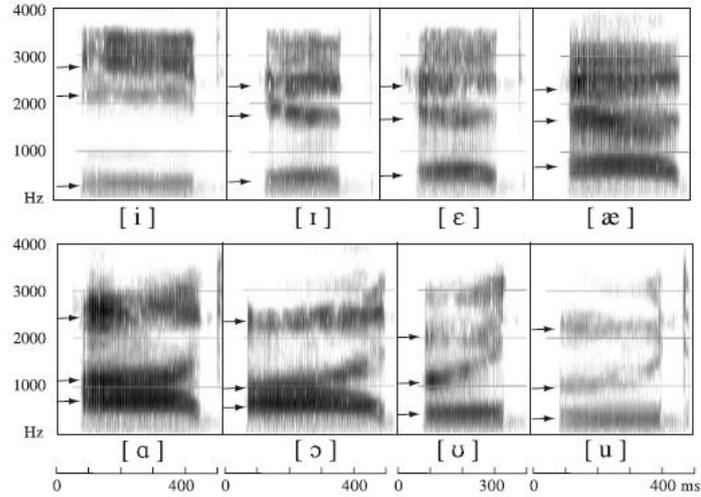
Figura 8 – Sinal de fala no domínio do tempo e de frequência.



Fonte: Dutoit e Marques (2010).

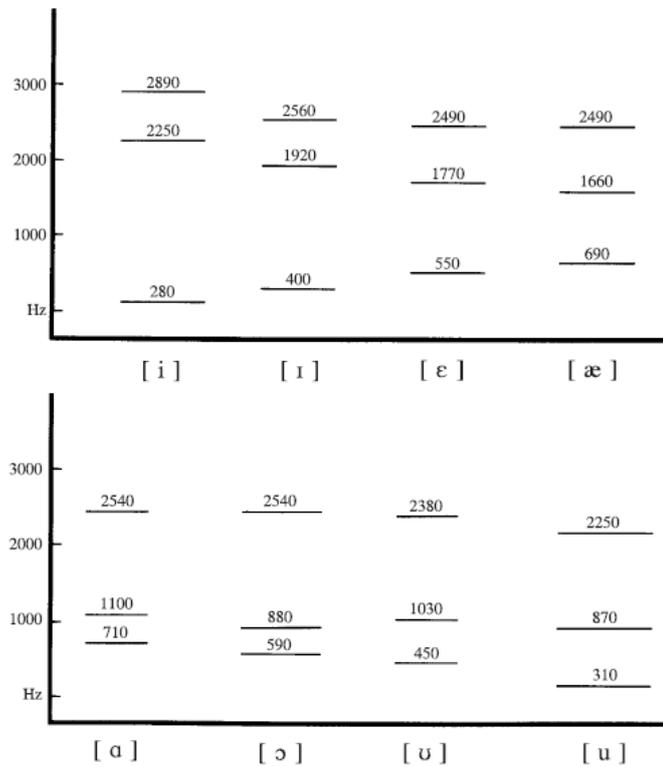
Na Figura 8, a frequência fundamental, denominada  $F_0$ , e suas harmônicas,  $H_{1,2,3,\dots,n}$ , que são características da excitação da glote, seguem presentes no sinal. Ocorre, entretanto, a modulação deste sinal, dando a ele a característica das frequências de ressonância do trato vocal, identificadas na figura como  $F_1, F_2, F_3$  e  $F_4$ . Estas frequências são denominadas frequências formantes e são elas que caracterizam a pronúncia de diferentes vogais (LADEFOGED; JOHNSON, 2010). A Figura 9 mostra o espectrograma da pronúncia de diferentes palavras por e a Figura 10 destaca as três primeiras formantes obtidas em cada uma das pronúncias.

Figura 9 – Espectrograma das palavras *heed, hid, head, had, hod, hawed, hood, who'd* pronunciadas por um americano do sexo masculino.



Fonte: Ladefoged e Johnson (2010).

Figura 10 – Três primeiras formantes das vogais pronunciadas na Figura 9.



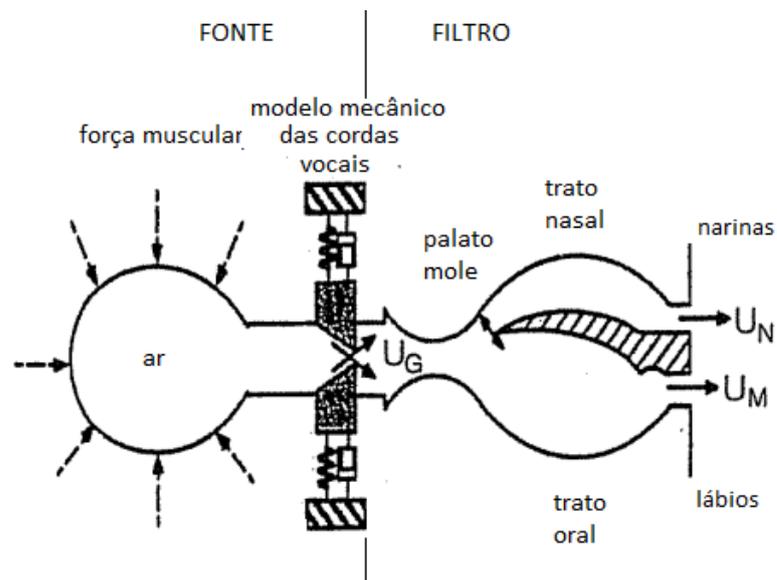
Fonte: Ladefoged e Johnson (2010).

Uma simples inspeção visual do espectro da fala mostrado na Figura 8 remete a um sinal periódico, ou *quasi periódico*, convoluído com um filtro cuja resposta em frequência corresponda ao envelope das frequências formantes.

A geração de sons não vozeados ocorre sem a utilização de cordas vocais, resultando unicamente da passagem de ar e contrações do trato vocal. Do ponto de vista de modelagem, o sinal de pressão sonora para sons não vozeados pode ser comparado a uma fonte de ruído.

O sistema fonador humano pode ser reproduzido por um modelo fonte-filtro, composto por duas etapas desacopladas: a excitação na frequência fundamental da fala, quando vozeada, ou ruído, quando não vozeada, e o filtro, com a resposta no domínio da frequência do trato vocal (FANT, 1971). Este modelo é mostrado na Figura 11, onde  $U_G$  é o sinal glotal e  $U_G$  e  $U_M$  são, respectivamente os sinais de fala oral e nasal.

Figura 11 – Modelo fonte-filtro.



Fonte: Modificado de Childers (2000).

### 2.1.2 Sintetização da fala humana - VOCODER

O processo de síntese de fala pode ser dividido em duas partes desacopladas: a excitação e a filtragem, seguindo o modelo fonte-filtro proposto por Gunnar Fant em 1960 (DUTOIT; MARQUES, 2010). A sintetização da fala consiste na reconstrução artificial deste modelo.

Makhoul (1975) define a fala como um processo estocástico localmente estacionário. Ou seja, quando analisada em intervalos definidos de tempo, observa-se um fenômeno estacionário. Logo, o processo de sintetização da fala deve ocorrer por janelas, dividindo o sinal em porções estacionárias. O tamanho da janela utilizada para análise da fala é tipicamente entre 20ms e 30ms (DUTOIT; MARQUES, 2010).

A sintetização exata da fala exige que seja identificado se o som da janela analisada é vozeado ou não vozeado e, a partir disto, definir o tipo de sinal que será utilizada como fonte na síntese. Para sons vozeados, identifica-se a frequência fundamental do *frame* e, na etapa de síntese, aplica-se um trem de impulsos delta de Kronecker<sup>1</sup> nessa frequência como sinal de excitação do sistema. Para síntese de janelas com som não vozeado, utiliza-se ruído branco como excitação (DUTOIT; MARQUES, 2010).

Por uma série de fatores, a determinação do tipo de som e frequência fundamental da janela analisada não é simples. O som analisado nunca é puramente vozeado ou não vozeado, o que dificulta a decisão do tipo de excitação que deve ser utilizado na síntese (DUTOIT; MARQUES 2010). Além disso, a variação da amplitude dos pulsos glotais e o fato deste sinal não ser exatamente periódico dificulta a estimação da frequência fundamental.

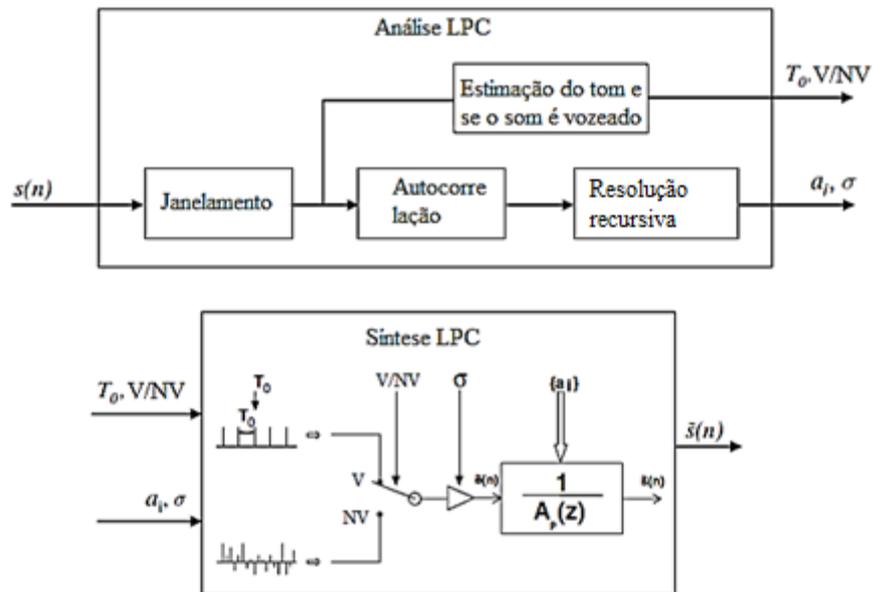
Um dos métodos mais simples de obtenção dos parâmetros da voz se dá através da transformada de Fourier do *frame* de fala, extraíndo o seu espectro. O espectro do *frame* analisado contém, em seu envelope, as frequências formantes da fala e, em sua estrutura mais fina, a frequência fundamental e suas harmônicas. A obtenção da frequência fundamental ocorre através da identificação da componente de frequência comum nas harmônicas da estrutura fina do espectro do *frame* e, a ausência desta componente comum, indica que o som é não vozeado. O método de análise por Fourier é denominado não-paramétrico, não seguindo um modelo específico e tendo aplicabilidade em qualquer tipo de sinal (FURUI, 2000).

Neste trabalho, é utilizada a codificação por predição linear, LPC. Este é um método do tipo paramétrico, pois segue o modelo de filtro *all-pole* para modelagem espectral da fala e extrai os parâmetros da voz através da autocorrelação ou covariância do sinal. O fato do LPC ser um modelo paramétrico, utilizando como base um modelo dedicado à análise de sinais de fala, o torna mais efetivo em relação à análise por Fourier (FURUI, 2000).

O diagrama de análise e síntese utilizando LPC é mostrado na Figura 12, onde as entradas  $T_0$ ,  $V/UV$ ,  $\sigma$  e  $a_i$  são, respectivamente, o período fundamental do *frame*, a definição

de vozeado ou não vozeado, o ganho e os coeficientes de predição. Os procedimentos deste método são detalhados na seção 2.2.

Figura 12 – Sistema de análise e síntese LPC.



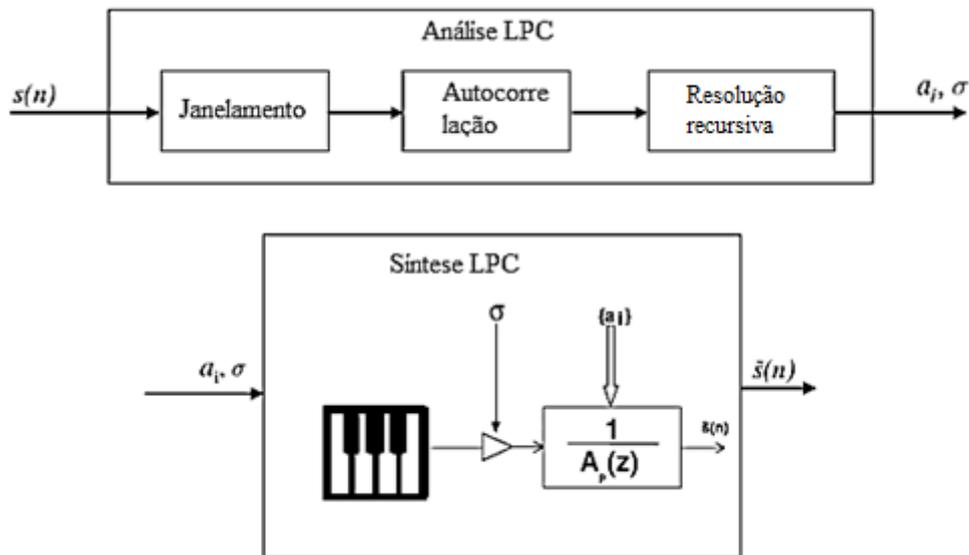
Fonte: Dutoit e Marques (2010).

### 2.1.3 Aplicação musical

Para aplicação musical, propõe-se a alteração da excitação do sistema, substituindo o trem de impulsos e a fonte de ruído por um instrumento musical. O efeito desejado no VOCODER musical é o de substituir, no sistema fonador humano, a excitação das cordas vocais por um instrumento musical, atribuindo ao som deste instrumento características do trato vocal do usuário. Pode-se manter a fonte de ruído para sons não vozeados, a fim de tornar mais nítidos estes sons, ou utilizar o instrumento como excitador geral.

Convenientemente, a definição de um instrumento musical como excitação geral do sistema elimina a necessidade de verificação do tipo de som da janela e do levantamento da frequência fundamental da janela analisada. Dessa forma, são necessários somente dois parâmetros para a síntese da fala: o ganho e os coeficientes de predição, ambos obtidos através da análise LPC, detalhada na seção 2.2.. Propõe-se então, para o sistema mostrado na Figura 12, uma construção mais simples, mostrada na Figura 13.

Figura 13 – Diagrama proposto de VOCODER musical.



## 2.2 O MÉTODO LPC – CODIFICAÇÃO POR PREDIÇÃO LINEAR

### 2.2.1 O conceito de predição linear

Predição linear (LPC – *Linear Prediction Coding*) é uma técnica amplamente utilizada no processamento da fala, que pode ser muito bem modelada através desta ferramenta.

A técnica de LPC consiste na análise da fala e, desta análise, a definição de parâmetros básicos para reconstrução dela de forma sintética: frequência fundamental, tipo de som (vozeado ou não vozeado), ganho e coeficientes de predição, que definem os polos do filtro LPC.

A predição linear parte do modelo que descreve o sinal como combinação linear de saídas e entradas passadas, conforme descreve a equação (1) (MAKHOUL, 1975)

$$x[n] = - \sum_{k=1}^p \alpha_k x[n-k] + G \sum_{i=1}^q b_i u[n-i], \quad (1)$$

onde  $\alpha_k$ ,  $b_i$  e o ganho  $G$  são os parâmetros que caracterizam o sistema,  $x[n]$  é o sinal de saída e  $u[n]$  o sinal de entrada. No domínio da frequência, fazendo a transformada  $z$  da equação (1), a função de transferência do sistema é descrita por

$$H(z) = G \frac{1 + \sum_{l=1}^q b_l z^{-l}}{1 + \sum_{k=1}^p \alpha_k z^{-k}} \quad (2)$$

A equação (2) descreve o preditor como um quociente de polinômios, onde as raízes do numerador e do denominador são, respectivamente, os zeros e os polos do sistema (HAYKIN; VAN VEEN, 2007).

Existem duas simplificações do modelo que são de interesse. Uma delas é o modelo *all-zero*, que zera todos os parâmetros  $\alpha_k$  e realiza a predição unicamente através das entradas passada. Outra é o modelo *all-pole*, que zera todos os parâmetros  $b_l$  e realiza a predição exclusivamente através das saídas passadas.

Neste trabalho, será utilizado o modelo *all-pole*, modelo no qual se baseia o método LPC.

### 2.2.2 O modelo *all-pole* de predição linear

Ao aplicar o modelo *all-pole* no sistema descrito na equação (1), este sistema passa a ser

$$\hat{x}[n] = - \sum_{k=1}^p \alpha_k x[n-k] + Gu[n] \quad , \quad (3)$$

onde  $\hat{x}[n]$  é a estimacão do sinal  $x[n]$  e  $p$  representa a quantidade de amostras passadas que serão utilizadas na predição, definindo também a ordem do filtro de predição a ser utilizado na sintetização,  $\alpha_k$  são os coeficientes utilizados no filtro de predição e  $G$  é o ganho da entrada desconhecida  $u[n]$ . No domínio  $z$ , pode-se escrever cada amostra como

$$\hat{X}(z) = \frac{G}{1 + \sum_{k=1}^p \alpha_k z^{-k}} \quad . \quad (4)$$

Ocorre que a entrada  $u[n]$ , na maioria das aplicações, não é conhecida, e, por isso, a predição do sinal  $x[n]$  se dá somente com a soma ponderada das amostras passadas. A estimacão de cada amostra passa a ser então

$$\hat{x}[n] = - \sum_{k=1}^p \alpha_k x[n-k] \quad . \quad (5)$$

### 2.2.3 Estimacão no domínio do tempo

A determinacão dos  $p$  coeficientes de predição pode se dar pela minimizacão do erro residual (MAKHOUL, 1975). O erro residual de predição é descrito por

$$e[n] = x[n] + \sum_{k=1}^p \alpha_k x[n-k] . \quad (6)$$

Um dos métodos para esta minimização se dá através de mínimos quadrados, que busca a minimização do erro quadrado total do sinal. Para um sinal determinístico, pode-se escrever o erro quadrado total como

$$E = \sum_n e[n]^2 = \sum_n \left( x[n] + \sum_{k=1}^p \alpha_k x[n-k] \right)^2 . \quad (7)$$

A minimização do erro quadrado total se dá então por

$$\frac{\partial E}{\partial \alpha_i} = 0, \quad 1 \leq i \leq p . \quad (8)$$

Obtém-se então o conjunto de  $p$  equações com  $p$  incógnitas, denominadas *equações normais*

$$\sum_{k=1}^p \alpha_k \sum_n x[n-k]x[n-i] = - \sum_n x[n]x[n-i], \quad 1 \leq i \leq p . \quad (9)$$

Assumindo uma minimização do erro ao longo de  $-\infty < n < \infty$ , a equação (9) pode ser reescrita em termos da autocorrelação do sinal, sendo reduzida à

$$\sum_{k=1}^p \alpha_k R(i-k) = -R(i), \quad 1 \leq i \leq p , \quad (10)$$

onde

$$R(i) = \sum_{n=-\infty}^{\infty} x[n]x[n+i] \quad (11)$$

é a autocorrelação do sinal  $x[n]$ .

Escrevendo a equação (10) em sua forma matricial, tem-se

$$\mathbf{R} \cdot \boldsymbol{\alpha} = -\mathbf{P}, \quad (12)$$

onde

$$\mathbf{R} = \begin{bmatrix} R_0 & R_1 & R_2 & \dots & R_{p-1} \\ R_1 & R_0 & R_1 & & R_{p-2} \\ R_2 & R_1 & R_0 & & R_{p-3} \\ \vdots & & & & \\ R_{p-1} & R_{p-2} & R_{p-3} & & R_0 \end{bmatrix}, \quad (13)$$

$$\boldsymbol{\alpha} = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \alpha_3 \\ \vdots \\ \alpha_p \end{bmatrix}, \quad (14)$$

$$\mathbf{P} = \begin{bmatrix} R_1 \\ R_2 \\ R_3 \\ \vdots \\ R_p \end{bmatrix}. \quad (15)$$

A obtenção dos parâmetros, então, se dá por

$$\boldsymbol{\alpha} = -\mathbf{R}^{-1} \cdot \mathbf{P}. \quad (16)$$

A obtenção dos parâmetros através da resolução da equação (16) é denominada método de Yule-Walker. Dependendo da ordem de predição, a inversão da matriz  $\mathbf{R}$  pode representar um custo computacional muito alto para aplicação direta do método de Yule-Walker. Aplica-se então, métodos de resolução recursivos para solução da equação (16). Um destes métodos é o método de Burg, que aborda o problema através da estimação da máxima entropia do espectro de potência do sinal. Para sistemas de alta ordem, entretanto, o método de Burg introduz picos espúrios no sinal, não sendo recomendado para o LPC (PROAKIS; MANOLAKIS, 2007). Outro método possível é o método de Schur, que é muito aplicado em computação paralela, não fazendo parte do escopo do presente trabalho. Por apresentar um bom resultado em sistemas de maior ordem e ser usual na análise LPC, o método escolhido para obtenção dos parâmetros de predição é recursão de Levinson-Durbin, explicada na seção 2.2.6.

Por ser processado por janelas, o sinal  $x[n]$  é conhecido num intervalo finito de tempo, uma vez que seus valores são nulos fora da janela analisada. Assim, para uma janela de tamanho  $N$ , que considera o sinal no intervalo  $0 \leq n \leq N - 1$ , a função de autocorrelação é dada por

$$R(i) = \sum_{n=0}^{N-1-i} x'[n]x'[n+i], \quad (17)$$

onde  $x'[n]$  é o sinal  $x[n]$  multiplicado por uma função de janelamento qualquer  $w[n]$ :

$$x'[n] = \begin{cases} x[n]w[n], & 0 \leq n \leq N - 1 \\ 0, & \text{caso contrário} \end{cases} \quad (18)$$

Ocorre, entretanto, que a fala não é um sinal determinístico, exigindo que se trabalhe com a minimização do valor esperado do erro quadrático (MAKHOUL, 1975). Dessa forma, para sinais randômicos, as equações (7) e (9), tem como equivalentes as equações

$$E = \mathcal{E}(e[n]^2) = \mathcal{E} \left( x[n] + \sum_{k=1}^p \alpha_k x[n-k] \right)^2, \quad (19)$$

$$\sum_{k=1}^p \alpha_k \mathcal{E}(x[n-k]x[n-i]) = -\mathcal{E}(x[n]x[n-i]), \quad 1 \leq i \leq p, \quad (20)$$

onde  $\mathcal{E}(e[n]^2)$  representa o valor esperado de  $e[n]^2$ .

Para processos estacionários, que é o caso de um sinal janelado, tem-se que:

$$\mathcal{E}(x[n-k]x[n-i]) = R(i-k). \quad (21)$$

onde  $R(i-k)$  é função autocorrelação do sinal de entrada. A equação (17) pode então ser reduzida à equação (10) com a única diferença de que a autocorrelação passa a ser de um sinal estacionário e não mais um sinal determinístico.

#### 2.2.4 Estimação no domínio da frequência – estimação espectral

A predição linear é basicamente uma análise de correlação, que pode ser realizada tanto no domínio do tempo quanto no domínio da frequência (MAKHOUL, 1975). Nesta seção, é mostrado o procedimento de predição através da análise no domínio da frequência.

Aplicando a transformada  $z$  na equação (6) que descreve o erro entre o sinal real e o predito, obtêm-se

$$E(z) = \left[ 1 + \sum_{n=-\infty}^{\infty} \alpha_n z^{-n} \right] X(z) = A(z)X(z). \quad (22)$$

$A(z)$  é denominado o filtro inverso de predição, enquanto  $E(z)$  e  $X(z)$  são, respectivamente, a transformada  $z$  do resíduo  $e[n]$  e do sinal  $x[n]$ . A equação (22) indica que o resíduo da predição é na verdade resultado da passagem do sinal  $X(z)$  pelo filtro inverso  $A(z)$ .

Considerando um sinal determinístico ou, como é o caso da presente aplicação, estacionário, e aplicando o teorema de Parseval, o erro total a ser minimizado é dado por

$$E = \sum_{n=-\infty}^{\infty} e[n]^2 = \frac{1}{2\pi} \int_{-\pi}^{\pi} |E(e^{j\omega})|^2 d\omega, \quad (23)$$

onde  $E(e^{j\omega})$  é obtido analisando  $E(z)$  no círculo unitário  $z = e^{j\omega}$ .

Definindo a densidade espectral do sinal  $x[n]$  como  $P(\omega)$ , onde

$$P(\omega) = |S(e^{j\omega})|^2, \quad (24)$$

o erro total a ser minimizado pode ser escrito como

$$E = \frac{1}{2\pi} \int_{-\pi}^{\pi} P(\omega) A(e^{j\omega}) A(e^{-j\omega}) d\omega. \quad (25)$$

Seguindo o mesmo procedimento da seção 2.2.3, o erro pode então ser minimizado aplicando a equação (8) à equação (25).

A autocorrelação do sinal pode ser obtida através da transformada inversa da densidade espectral do sinal:

$$R(i) = \frac{1}{2\pi} \int_{-\pi}^{\pi} P(\omega) \cos(i\omega) d\omega \quad (26)$$

### 2.2.5 Determinação da ordem de predição

A ordem do preditor,  $p$ , que também é a quantidade de polos no filtro de síntese, deve ser escolhida de forma que a síntese resultante tenha graus de liberdade suficientes para copiar o envelope espectral do sinal de entrada. Como existe, aproximadamente, um formante por kHz da largura de banda do sinal de fala, são necessários ao menos  $2B$  polos, onde  $B$  é a largura de banda do sinal de fala em kHz (DUTOIT; MARQUES, 2010). Empiricamente, são adicionados dois polos ao valor mínimo (DUTOIT; MARQUES, 2010).

Dessa forma, considerando uma banda de 4 kHz, como é o caso do sistema testado no presente trabalho, a ordem de predição mínima é  $p=8$ . Adicionando dois polos a esse valor, tem-se uma predição de ordem 10.

### 2.2.6 Obtenção dos parâmetros pelo Algoritmo de Levinson-Durbin

O método de Levinson-Durbin trabalha com a solução a partir da matriz de autocorrelação e tem um custo computacional de  $p^2 + O(p)$  operações, onde  $p$  é a quantidade de coeficientes e  $O(p)$  é o fator de máximo crescimento da função.

Na equação (12), a matriz de autocorrelação  $\mathbf{R}$ , de tamanho  $p \times p$ , é simétrica e os elementos ao longo de qualquer diagonal são idênticos. Matrizes com essa característica são denominadas matrizes de Toeplitz (MAKHOUL, 1975). Além disso, o método se vale do fato de que o vetor coluna  $\mathbf{P}$ , do lado direito da equação (12), contém os mesmo elementos presentes na matriz de autocorrelação.

O método de Levinson-Durbin trabalha com a minimização do erro e utiliza como condição inicial  $E_0 = R(0)$ , trabalhando recursivamente, com  $i = 1, 2, \dots, p$ , nas equações a seguir.

$$k_i = - \left[ R(i) + \sum_{j=1}^{i-1} \alpha_j^{(i-1)} R(i-j) \right] / E_{i-1}, \quad (27a)$$

$$\alpha_i^{(i)} = k_i, \quad (27b)$$

$$\alpha_j^{(i)} = \alpha_j^{(i-1)} + k_i \alpha_{i-j}^{(i-1)}, \quad 1 \leq j \leq i-1, \quad (27c)$$

$$E_i = (1 - k_i^2) E_{i-1}. \quad (27d)$$

Após as  $p$  iterações, a solução final é dada por

$$\alpha_j = \alpha_j^{(p)}, \quad 1 \leq j \leq p. \quad (28)$$

## 2.3 EFEITOS DA PRECISÃO NUMÉRICA LIMITADA EM SISTEMAS DIGITAIS

### 2.3.1 Representação de números em ponto fixo

A representação de números em ponto fixo recebe este nome por ter uma faixa fixa de números que podem representar um determinado valor, com uma posição predeterminada da vírgula. Assim, para uma determinada operação, todos os valores representados em ponto fixo possuem uma mesma quantidade definida de algarismos inteiros e de algarismos fracionários.

A representação de um determinado número  $X$  em ponto fixo se dá por

$$X = (b_{-(A-1)} \dots b_{-1}, b_0, b_1, \dots, b_B) = \sum_{i=-(A-1)}^B b_i r^{-i}, \quad 0 \leq b_i \leq (r-1), \quad (29)$$

onde  $b_i$  representa o dígito,  $r$  representa a base,  $A$  o número de dígitos inteiros e  $B$  o número de dígitos fracionários.

Para processamento digital de sinais, a representação binária é a de interesse. Nesse caso temos que  $r = 2$  e os dígitos  $b_i$  são chamados de *bits*, podendo assumir os valores 0 ou 1. O dígito binário  $b_{-(A-1)}$  é chamado de dígito mais significativo (MSB) e o dígito  $b_B$  é chamado de dígito menos significativo (LSB).

Na representação de um número inteiro positivo com  $n$  bits ( $A = n - 1, B = 0$ ), pode-se representar números inteiros positivos dentro de uma faixa de 0 a  $2^n - 1$ . Para números

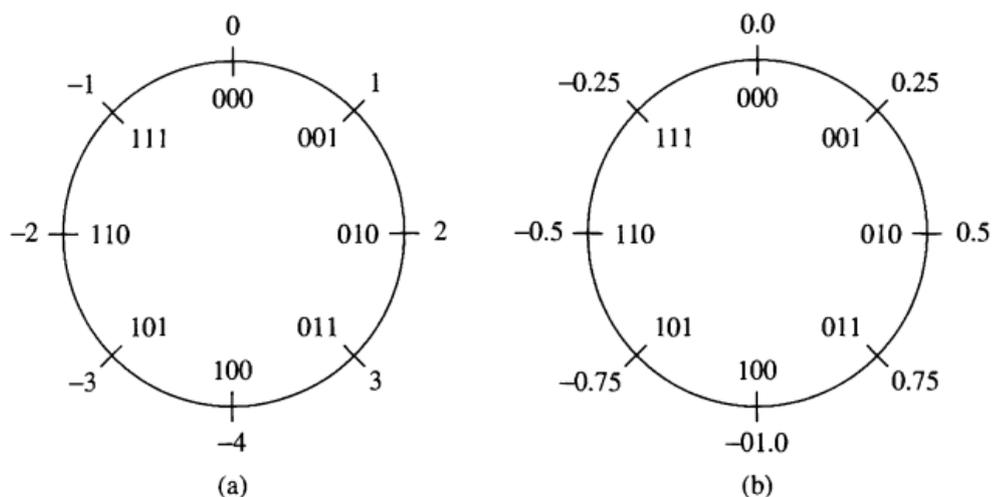
fracionários positivos ( $A = 0$ ,  $B = n - 1$ ), utiliza-se um ponto binário entre  $b_0$  e  $b_1$ , o que define a faixa de números representáveis, para  $n$  bits, de  $0$  a  $1 - 2^{-n}$ .

O ponto decimal entre os dígitos  $b_0$  e  $b_1$  não existe fisicamente em sistemas digitais. Simplesmente, os circuitos lógicos de um computador são projetados de forma que todos os cálculos resultem em números que correspondam à posição assumida do ponto decimal (PROAKIS; MANOLAKIS, 2007).

Existem três possíveis formas de representação binária de números negativos: a reserva de um bit para indicação de sinal, o método de complemento de um e o método de complemento de dois.

A representação de números negativos mais utilizado por processadores com arquitetura de ponto fixo é o complemento de dois. Neste método, um número negativo é representado invertendo o valor dos seus dígitos e, após esta inversão, somando 1 ao seu bit menos significativo. O método é ilustrado na Figura 8, que mostra o círculo de complemento de dois para um número de três bits inteiro e fracionário. Nesta representação, a faixa de representação para números de  $(B+1)$  bits vai de  $-1$  a  $1 - 2^{-B}$  para frações e de  $-2^B$  a  $2^B - 1$  para inteiros.

Figura 14 – Círculo de complemento de dois para uma número de 3 bits (a) inteiro e (b) fracionário.



Fonte: Proakis e Manolakis (2007).

Uma propriedade muito importante da aritmética de complemento de dois é que, se a soma de final de uma série de números  $X_1, X_2, \dots, X_N$  se encontra dentro da faixa de possível

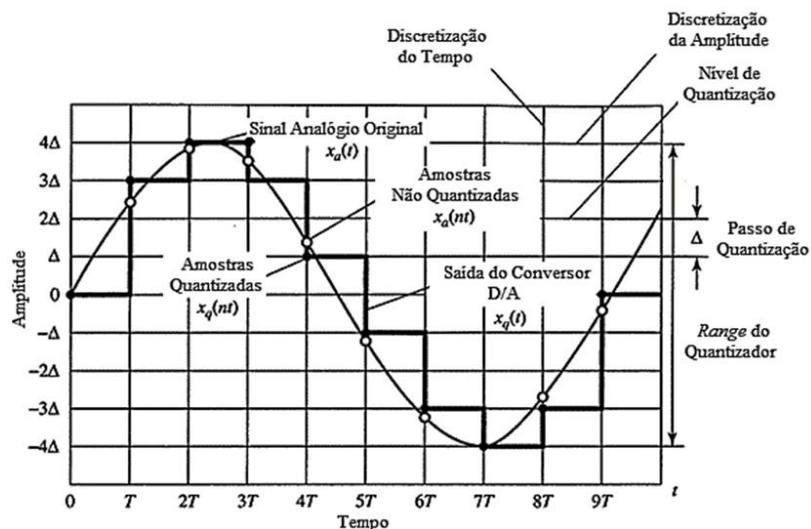
representação do tamanho da palavra, esta soma é computada corretamente, mesmo que, em meio ao processo, alguma das somas parciais tenha extrapolado esta faixa (PROAKIS; MANOLAKIS, 2007).

No caso de produtos de número representados em ponto fixo, a multiplicação de dois números de  $b$  bits resulta num produto com tamanho  $2b$  bits. Com isso, ocorre o truncamento ou arredondamento nos últimos  $b$  bits da multiplicação, contribuindo nos erros de quantização.

### 2.3.2 Quantização

A quantização é o processo de converter um sinal de amplitude contínua num sinal digital com um número finito de dígitos. Num sinal quantizado, o valor de cada amostra é representado por um valor selecionado de um conjunto de possíveis valores (PROAKIS; MANOLAKIS, 2007). A Figura 15 mostra a quantização de um sinal senoidal amostrado.

Figura 15 – Exemplo de quantização de sinal senoidal.



Fonte: Adaptado de Proakis e Manolakis (2007).

Em teoria, a quantização de sinais analógicos sempre acarreta na perda de informação, perda esta que resulta da ambiguidade introduzida pela quantização. A quantização é, de fato, um processo irreversível (PROAKIS; MANOLAKIS, 2007).

A precisão de um sistema, quando quantizado, depende do tamanho de seu passo de quantização  $\Delta$  e, em sistemas digitais, é definido por

$$\Delta = \frac{2X_m}{2^{B+1}} = \frac{X_m}{2^B}, \quad (30)$$

onde  $X_m$  é o valor máximo da escala do conversor A/D e B é o tamanho de palavra do código binário que representa o valor quantizado. O passo de quantização corresponde ao bit menos significativo do número binário representando a amostra quantizada (OPPENHEIM; SCCHAFER, 1999).

Denota-se erro de quantização, ou ruído de quantização, o erro introduzido pela quantização de um sinal. Este erro é representado por

$$e_q[n] = x_q[n] - x[n], \quad (31)$$

onde  $x_q(n)$  é o sinal  $x(n)$  quantizado.

No caso geral de um quantizador de (B + 1) bits o erro de quantização satisfaz a equação

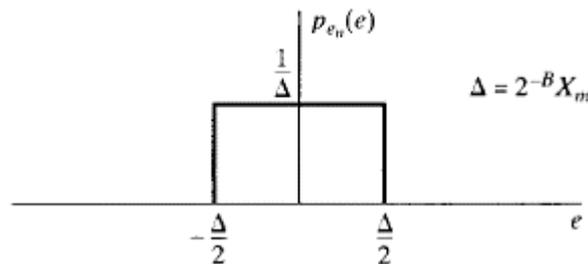
$$-\frac{\Delta}{2} < e_q[n] \leq \frac{\Delta}{2}, \quad (32)$$

quando o sinal  $x[n]$  satisfaz

$$\left(-X_m - \frac{\Delta}{2}\right) < x[n] < \left(X_m - \frac{\Delta}{2}\right). \quad (33)$$

Para um passo de quantização pequeno, pode-se assumir que o ruído de quantização tem uma variação randômica uniformemente distribuída de  $-\Delta/2$  a  $\Delta/2$ , conforme mostra a Figura 16 (OPPENHEIM, SCHAFFER, 1999).

Figura 16 – Distribuição de probabilidade do erro de quantização.



Fonte: Oppenheim e Schafer (1999).

O valor médio de  $e[n]$  é zero, e sua variância é

$$\sigma_e^2 = \int_{-\frac{\Delta}{2}}^{\frac{\Delta}{2}} e^2 \frac{1}{\Delta} de = \frac{\Delta^2}{12}. \quad (34)$$

Para um quantizador de  $(B+1)$  bit com um valor máximo de escala  $X_m$ , a variância do ruído de quantização é dado por

$$\sigma_e^2 = \frac{2^{-2B} X_m^2}{12}. \quad (35)$$

Uma forma de medir a degradação do sinal pelo acréscimo de ruído é através da relação sinal-ruído, cuja equação se dá por

$$SNR = 10 \log_{10} \left( \frac{\sigma_x^2}{\sigma_e^2} \right) = 10 \log_{10} \left( \frac{12 \times 2^{2B} \sigma_x^2}{X_m^2} \right) = 6,02B + 10,8 - 20 \log_{10} \left( \frac{X_m}{\sigma_x} \right), \quad (36)$$

indicando que a relação sinal-ruído aumenta em aproximadamente 6 dB para cada bit adicionado ao tamanho de palavra do quantizador.

### 2.3.3 Quantização de coeficientes de filtros digitais

Na utilização de filtros digitais FIR ou IIR em computadores e sistemas digitais em geral, a precisão com que os coeficientes de tais filtros pode ser especificada é limitada pelo tamanho de palavra da máquina ou do registrador onde estes coeficientes são armazenados (PROAKIS; MANOLAKIS, 2007). Esta limitação faz com que, em geral, os polos e zeros dos filtros concebidos em sistemas digitais sejam diferentes do desejado, ocasionando, por consequência, numa resposta em frequência também diferente.

Dado um filtro IIR genérico representado por

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{1 + \sum_{k=1}^n \alpha_k z^{-k}}, \quad (37)$$

sua aplicação com coeficientes quantizados é escrita como

$$\bar{H}(z) = \frac{\sum_{k=0}^M \bar{b}_k z^{-k}}{1 + \sum_{k=1}^n \bar{\alpha}_k z^{-k}}, \quad (38)$$

onde os coeficientes quantizados podem ser escritos como a soma de seus valores originais com os erros de quantização  $\Delta\alpha_k$  e  $\Delta b_k$ .

$$\begin{aligned} \bar{\alpha}_k &= \alpha_k + \Delta\alpha_k, \quad k = 1, 2, \dots, N; \\ \bar{b}_k &= b_k + \Delta b_k, \quad k = 1, 2, \dots, N. \end{aligned} \quad (39)$$

O denominador de  $H(z)$  pode ser expresso na forma

$$D(z) = 1 + \sum_{k=0}^N \alpha_k z^{-k} = \prod_{k=1}^N (1 - p_k z^{-1}), \quad (40)$$

onde  $p_k$  são os polos de  $H(z)$ . No caso quantizado, os polos de  $\bar{H}(z)$  podem ser expressos como

$$\bar{D}(z) = \prod_{k=1}^N (1 - \bar{p}_k z^{-1}) , \quad (41)$$

com

$$\bar{p}_k = p_k + \Delta p_k , \quad k = 1, 2, \dots, N , \quad (42)$$

onde  $\Delta p_k$  é a perturbação relacionada à quantização de  $a_k$ .

O erro de perturbação  $\Delta p_i$  pode ser expresso como

$$\Delta p_i = \sum_{k=1}^N \frac{\partial p_i}{\partial \alpha_k} \Delta \alpha_k , \quad (43)$$

onde a derivada parcial de  $p_i$  em relação à  $\alpha_k$  representa a mudança incremental no polo  $p_i$  em relação a mudanças no coeficiente  $\alpha_k$  e pode ser expressa como

$$\frac{\partial p_i}{\partial \alpha_k} = \frac{-p_i^{N-k}}{\prod_{\substack{l=1 \\ l \neq i}}^N (p_i - p_l)} . \quad (44)$$

Substituindo a equação (44) na equação (43), o erro total de perturbação pode ser escrito como

$$\Delta p_i = - \sum_{k=1}^N \frac{p_i^{N-k}}{\prod_{\substack{l=1 \\ l \neq i}}^N (p_i - p_l)} \Delta \alpha_k . \quad (45)$$

Através da expressão (45), obtém-se uma medida da sensibilidade do  $i$ -ésimo polo às mudanças nos coeficientes  $\alpha_k$ . Resultados análogos podem ser obtidos quando o mesmo procedimento matemático é aplicado aos parâmetros  $b_k$ .

Os termos  $(p_i - p_l)$  no denominador da expressão (45) representam vetores no plano- $z$  dos polos  $p_l$  ao polo  $p_i$ . Se os polos se acumulam numa pequena região do plano- $z$ , como ocorre em filtros de banda estreita, a distância  $|p_i - p_l|$  se torna pequena e contribui para o aumento do erro de perturbação.

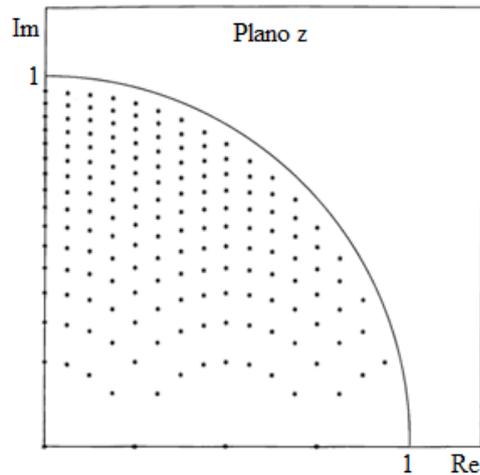
Um dos efeitos práticos da quantização de coeficientes de filtros IIR é a limitação nas posições possíveis dos polos no plano  $z$ . Por exemplo, considerando um filtro com dois polos, com a função de transferência descrita pela equação

$$H(z) = \frac{1}{1 - (2r \cos \theta) z^{-1} + r^2 z^{-2}} , \quad (46)$$

sem a quantização dos parâmetros, os polos podem assumir infinitas posições no plano  $z$ . Entretanto, limitando o tamanho de palavra dos coeficientes à 4 bits, o que implica em 15

possíveis valores não-nulos, o número de possíveis posições dos polos se limita a 169 por quadrante, conforme ilustra a Figura 17.

Figura 17 – Possíveis posições no plano  $z$  dos polos para o filtro da equação (55) quantizado com 4 bits.

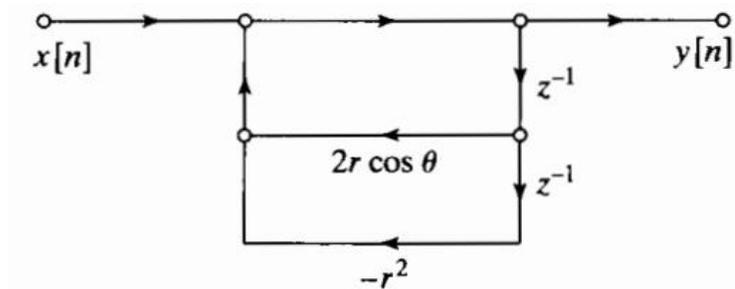


Fonte: Modificado de Proakis e Manolakis (2007).

A distribuição não uniforme dos polos na Figura 17 se deve à quantização do termo quadrático  $r^2$ . Esta distribuição representa uma situação desfavorável para filtros passa-baixa e passa-alta, que normalmente apresentam polos acumulados próximos dos eixos do plano  $z$ .

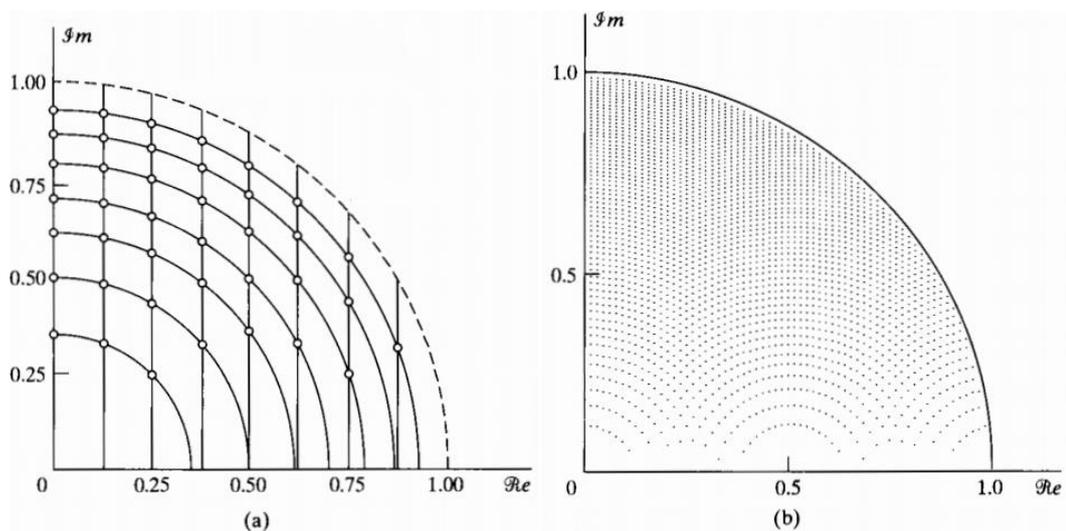
Considerando o sistema da Figura 18, a Figura 19 compara as possíveis posições dos polos no plano  $z$  para realizando uma quantização de 4 bits às possíveis posições dos polos do mesmo sistema mas com uma quantização de 7 bits.

Figura 18 – Implementação direta de par de polos complexos conjugados.



Fonte: Oppenheim e Schafer (1999).

Figura 19 – Posições possíveis dos polos do sistema mostrado na Figura 18 quantizado com (a) 4 bits e (b) 7 bits.



Fonte: Oppenheim e Schafer (1999).

Mudanças na posição dos polos podem levar à mudanças na resposta em frequência do sistema. A sensibilidade do sistema a variações na posição dos polos aumenta com a sua ordem. O sistema LPC aplicado no trabalho é de décima ordem, tornando crítica a precisão no posicionamento de seus polos. A questão de precisão na posição dos polos se torna ainda mais crítica no caso de polos posicionados próximos ao círculo unitário, uma vez que, por arredondamento, estes polos podem ser posicionados fora do círculo unitário, comprometendo a estabilidade do sistema.

Uma solução comum para minimizar a sensibilidade ao posicionamento dos polos em sistemas de alta ordem é a decomposição destes sistemas em subestruturas biquadráticas

cascadeadas (OPPENHEIM; SCHAFER, 1999). Esta solução, entretanto, exige o recálculo dos coeficientes do sistema, sendo mais custosa do ponto de vista computacional, fator que, para o LPC, impossibilita sua aplicação.

### 3 MATERIAIS E MÉTODOS

#### 3.1 MATERIAIS

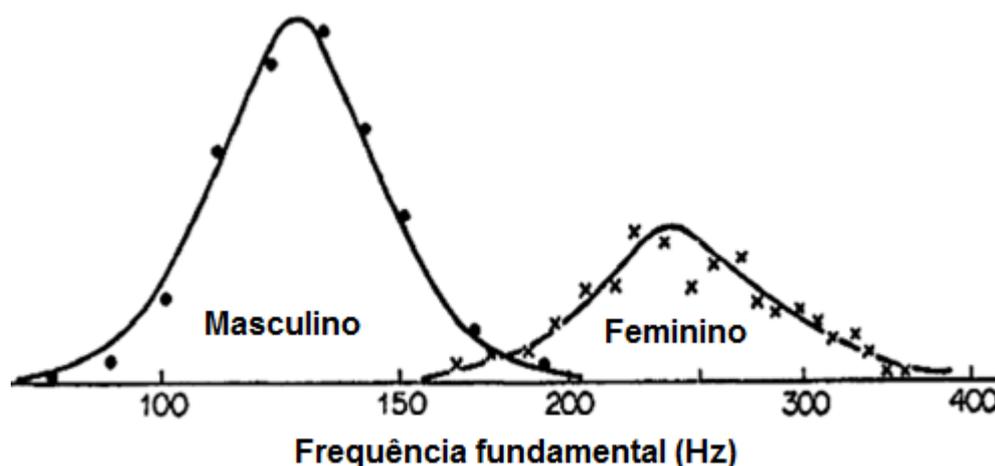
Nesta seção, são descritas as definições iniciais e ferramentas utilizadas para realização dos experimentos, que são descritos na seção 3.2.

##### 3.1.1 Sinal de fala e frequência de amostragem

A definição de frequência de amostragem da fala depende de dois fatores do sinal: a frequência fundamental e a posição das frequências formantes no espectro. Deve-se garantir uma frequência de amostragem que permita o levantamento destes parâmetros.

A Figura 20 mostra a distribuição estatística das frequências fundamentais para vozes femininas e masculinas. A figura evidencia que a frequência fundamental humana não ultrapassa 400 Hz.

Figura 20 – Distribuição da frequência fundamental da voz masculina e feminina.



Fonte: Furui (2000).

Em relação às frequências formantes, a Figura 8 deixa evidente que as 4 principais formantes da fala se encontram em frequências abaixo de 4 kHz.

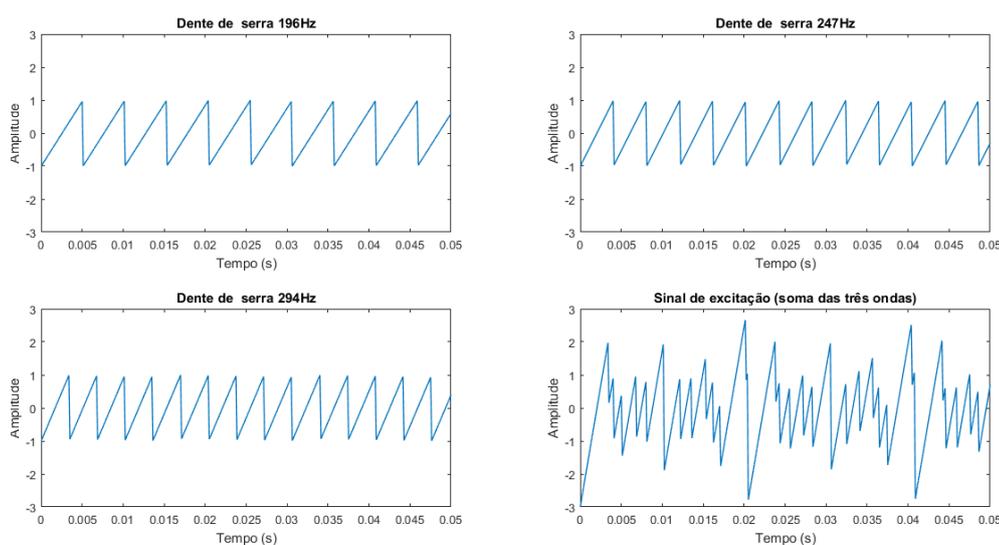
Pelo critério de Nyquist, a frequência de amostragem deve ser no mínimo o dobro da largura de banda do sinal (FURUI, 2000). Assim, considerando 4kHz como a máxima frequência de interesse, a frequência de amostragem utilizada no experimento foi de 8kHz. Essa frequência é a mesma utilizada para telefonia móvel (DUTOIT e MARQUES, 2010).

Para o experimento, foi utilizado um arquivo *.wav* com uma gravação de fala já numa frequência de amostragem em 8kHz e com um único canal de áudio, evitando assim a necessidade de manipulação do sinal. A leitura da gravação se deu através de função *audioread* do *MATLAB*.

### 3.1.2 Sinal de excitação musical do sistema

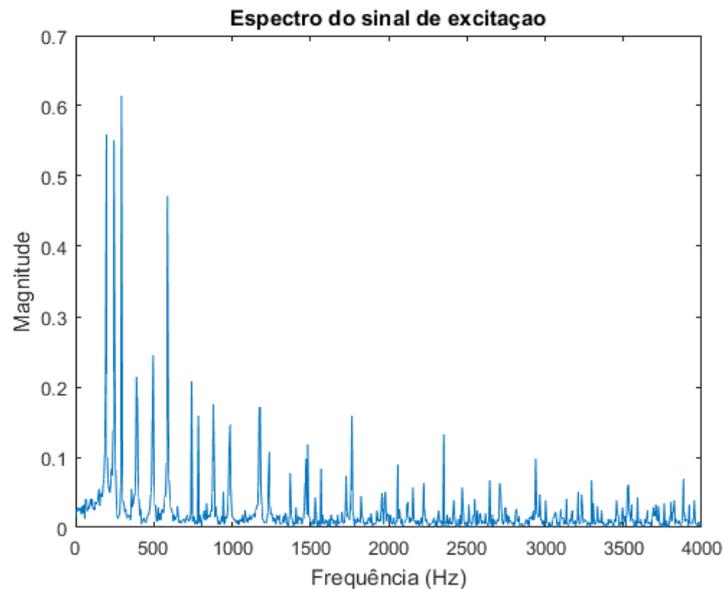
No VOCODER musical, a excitação do sistema é o sinal do instrumento musical. No experimento, utilizando a função *sawtooth* do *MATLAB*, o sinal de excitação foi composto pela soma de 3 ondas dente de serra, representando um acorde maior de um sintetizador. Para a formação do acorde de Sol Maior, utilizou-se frequências de aproximadamente 196 Hz, 247 Hz e 294 Hz. O gráfico do sinal de excitação é mostrado na Figura 21.

Figura 21 – Sinal de excitação do sistema.



Optou-se pelo formato dente de serra por representar um sinal rico em harmônicas, o que é uma característica dos sinais usados para excitação do VOCODER não musical. A Figura 22 mostra a resposta em frequência do sinal de excitação do sistema. Além disso, a onda dente-de-serra é largamente utilizada em sintetizadores como fonte de primordial para diversos efeitos.

Figura 22 – Espectro de frequência do sinal de excitação do sistema.



### 3.1.3 Janelamento e sobreposição

Uma vez que a fala é um sinal localmente estacionário, seu processamento deve ocorrer por janelas. No experimento, foi aplicada a janela de Hanning, cuja função se dá por

$$w(n) = 0.5 \left[ 1 - \cos \left( \frac{2\pi n}{N-1} \right) \right], \quad -\leq n \leq N, \quad (59)$$

onde o tamanho da janela, em amostras se dá por

$$L = N + 1 . \quad (60)$$

Optou-se por uma janela de 30 milissegundos para o processamento da fala. Para uma taxa de amostragem de 8kHz, 30 milissegundos equivalem a

$$L = t \times f_s = 0,03 \times 8000 = 240 \text{ amostras} . \quad (61)$$

Para posterior aplicação com o algoritmo *radix-2* no cálculo de transformada de Fourier discreta, definiu-se como comprimento da janela o valor mais próximo de 240 que fosse potência de 2, resultando em 256 amostras, levando a um janelamento de 32 milissegundos. As Figuras 17 e 18 mostram os gráficos da janela nos domínios do tempo e da frequência, gerados no *MATLAB*.

Figura 23 – Janela de Hanning aplicada (domínio do tempo).

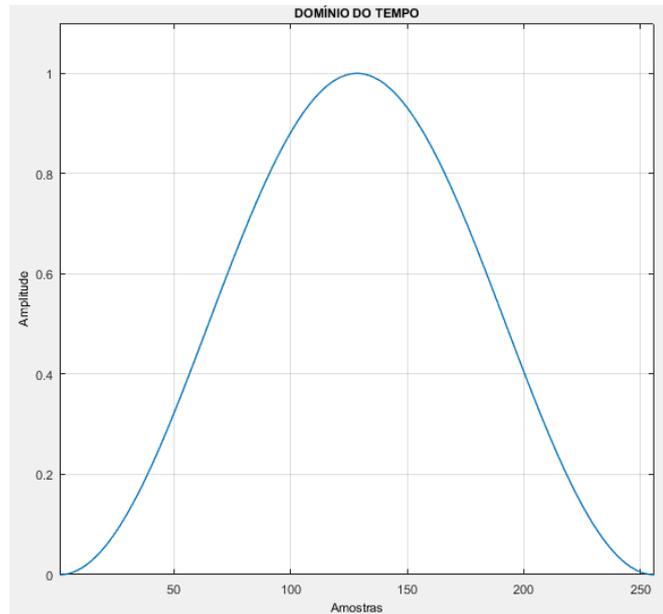
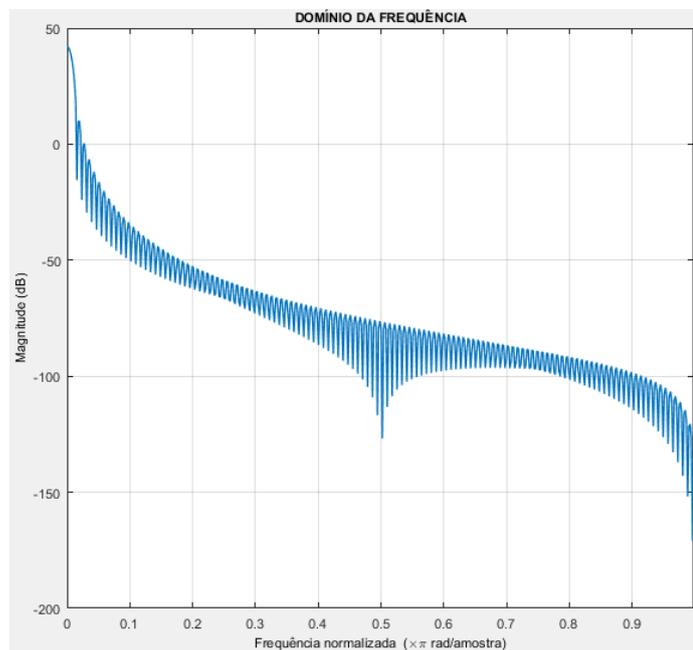
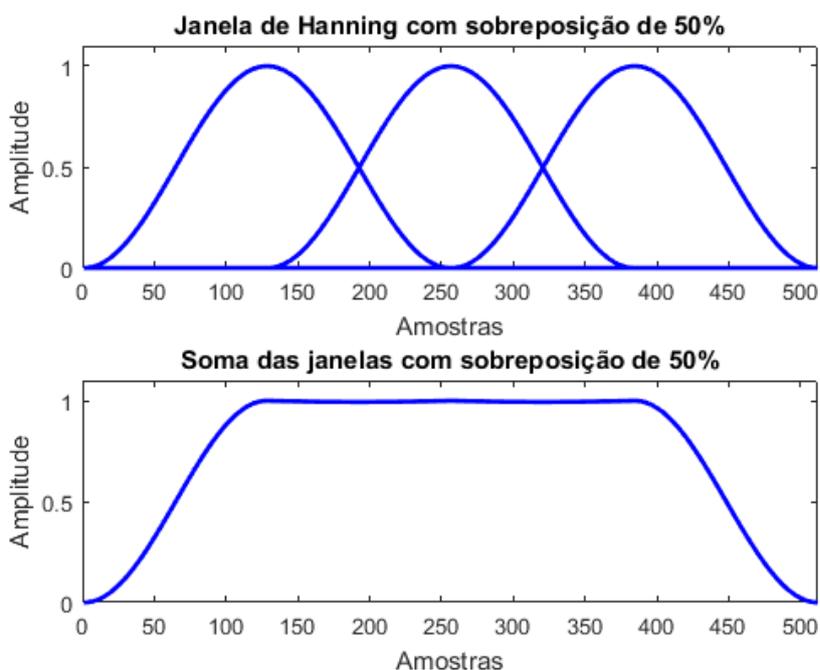


Figura 24 – Janela de Hanning aplicada (domínio da frequência).



O fator de sobreposição, também chamado de *overlap*, das janelas escolhido foi de 50% pois, para a janela de Hanning, esta sobreposição resulta em uma resposta praticamente plana, evitando que a janela interfira no peso dado a cada amostra do sinal.

Figura 25 – Sobreposição da janela de Hanning aplicada.



A conveniência de uma resposta plana com 50% de *overlap* é a característica que justificou a escolha da janela de Hanning.

### 3.1.4 Algoritmo LPC

O algoritmo LPC utilizado no experimento foi o da função *lpc.m*, do *MATLAB*. A função tem como entradas o sinal a ser analisado e a quantidade de coeficientes de predição desejados e, como saídas, os coeficientes de predição e a variância do erro de predição.

O algoritmo realiza a predição através da resolução da autocorrelação do sinal, resolvendo a equação (12). A análise do sinal é feita no domínio da frequência, através de estimação espectral, método explicado na seção 2.2.3. A resolução da matriz de autocorrelação se dá através da recursão de Levinson-Durbin, detalhada na seção 2.2.6.

Para o experimento, utilizou-se 10 coeficientes de predição na análise LPC.

### 3.1.5 Síntese LPC

De posse dos coeficientes LPC adquiridos da análise LPC, a síntese ocorre através da função *filter* do *MATLAB*. Os parâmetros de entrada são o numerador do filtro, o

denominador do filtro e o sinal a ser filtrado. A função retorna o sinal de entrada filtrado pelo filtro digital criado com o denominador e numerador de entrada.

Para o experimento, definiu-se como numerador o desvio padrão do erro de predição, proveniente da raiz da variância do resíduo, valor retornado pela função *lpc.m*. Trata-se, na verdade, de um ganho que varia de acordo com a energia do resíduo da predição do *frame* e tem como função manter uma relação fixa entre a energia de entrada e a energia de saída do sistema.

O denominador do filtro é formado pelos coeficientes LPC e o sinal a ser filtrado é o acorde musical da Figura 21.

### 3.1.6 Quantização dos sinais de entrada

A quantização dos sinais se deu através da função *quantize*, do *MATLAB*. Nesta função, os parâmetros de entrada são: a variável, o tipo numérico, o método de arredondamento e o tipo de ação tomada com o overflow.

O tipo numérico é definido por: sinalizado ou não, quantidade total de bits e quantidade de bits dedicada à parte fracionária. Para o experimento, manteve-se somente 1 bit dedicado à parte inteira do sinal, reservando o resto para representação da parte fracionária. Além disso, como os sinais excursionam tanto para valores positivos quanto negativos, o tipo numérico aplicado foi sempre do tipo sinalizado. O método de arredondamento aplicado foi do tipo truncamento e a ação para overflow definida como saturação do sinal. Antes da quantização, converteu-se as variáveis para ponto fixo, utilizando a função *fi* do *MATLAB*.

### 3.1.7 Modificação do algoritmo de análise LPC para funcionamento com ponto-fixo.

A função *lpc.m* do *MATLAB* não trabalha com ponto fixo. Dessa forma, foi necessária a modificação parcial da mesma. A principal mudança foi a substituição do algoritmo de transformada rápida de Fourier padrão do *MATLAB* pelo algoritmo de Cooley-Turkey, também conhecido por *radix2*, que faz a transformada por decimação no tempo.

Como se espera que os efeitos críticos da quantização ocorram na alocação dos polos do filtro LPC, restringiu-se a modificação do algoritmo à parte de análise, convertendo as variáveis para ponto flutuante no momento da passagem do sinal de excitação pelo filtro resultando de análise LPC.

## 3.2 MÉTODOS

Nesta seção, são descritos os experimentos realizados no trabalho. Todos os resultados são mostrados no capítulo 4.

### 3.2.1 Prova de conceito de LPC para aplicação musical

Nesta etapa do experimento não foi realizada a quantização do sinal. Todos os códigos foram implementados com variáveis do tipo *double*, representando uma precisão não reprodutível em sistemas digitais de menor custo.

#### 3.2.1.1 Análise de frame de fala isolado

Neste experimento, obteve-se o filtro LPC de um frame de fala isolado.

Para isso, através da função *audioread* do *MATLAB*, carregou-se um arquivo com uma gravação de fala *.wav*. Separou-se, então, um trecho de 256 amostras deste sinal de fala, cujos gráficos de amplitude e densidade espectral são mostrados na Figura 26 e na Figura 27.

De posse do *frame* isolado de fala, foi realizada a análise LPC dele, utilizando a função *lpc.m* do *MATLAB*. Realizou-se uma análise de ordem 10, conforme é explicado a seção 2.2.5. Os coeficientes de predição resultantes da análise são mostrados na Tabela 1.

De posse dos coeficientes da análise LPC, utilizando as funções *freqz* e *zplan*, plotou-se, respectivamente, a resposta em frequência do filtro LPC e a alocação dos polos deste filtro no plano *z*, mostrados na Figura 28 e na Figura 29.

Por fim, sobrepôs-se a resposta em frequência do filtro à densidade espectral da fala, para verificar se o formato da resposta do filtro acompanhava o envelope das frequências formantes da fala. Esta sobreposição é mostrada na Figura 30.

#### 3.2.1.2 *Benchmark* do filtro LPC obtido

Este experimento teve como objetivo a avaliação do algoritmo LPC de ordem 10.

De posse do filtro LPC obtido no experimento anterior, aplicou-se um trem de impulsos na frequência fundamental da fala no *frame* que originou o filtro. Buscou-se aqui a síntese exata do sinal de fala para posterior comparação com o sinal original.

Para isso, através da função *filter*, aplicou-se um trem de impulsos na frequência fundamental do *frame* de fala analisado anteriormente no filtro LPC, buscando assim a síntese do sinal de fala. A Figura 31 mostra o trem de impulsos aplicado como fonte de excitação para sintetização do *frame*.

A comparação entre o sinal sintetizado e o sinal original é mostrada na Figura 32 e na Figura 33.

#### 3.2.1.3 Síntese de *frame* único usando excitação musical.

Nesta etapa, analisou-se a síntese utilizando como excitação o sinal da Figura 21, proposto para validação musical do método LPC. O procedimento nesta etapa foi semelhante ao da seção 3.2.1.2 servindo, principalmente, como passo intermediário para a sintetização do sinal de fala inteiro com excitação musical.

A Figura 34 mostra a densidade espectral do sinal de excitação e a Figura 35 mostra a densidade espectral do sinal sintetizado.

#### 3.2.1.4 Análise e síntese musical com sinal de fala inteiro.

Nesta etapa, realizou-se o janelamento e *overlap* do sinal inteiro de fala e excitação, realizando a análise e síntese em cada *frame* e acumulando o sinal sintetizado num *buffer*. Foi aplicada uma janela de Hanning com 256 pontos e um *overlap* de 50%. Para realizar a análise e síntese LPC do sinal inteiro de fala, com uma duração de 6,37 segundos, foram necessárias 396 iterações, resultando em 396 análises e sínteses LPC. Após o sinal inteiro ter sido sintetizado, reproduziu-se o resultado utilizando a função *sound* do *MATLAB*.

A única análise quantitativa desta etapa foi a verificação de estabilidade dos filtros LPC ao longo da análise do sinal inteiro. Para isso, plotou-se, no plano *z*, os polos de todos os 396 filtros gerados ao longo da análise do sinal inteiro, para que fosse possível verificar a existência de polos fora do círculo unitário. A sobreposição dos polos no primeiro quadrante do plano *z* é mostrado na Figura 36, no segundo quadrante na Figura 37, no terceiro quadrante na Figura 38 e no quarto quadrante na Figura 39.

A análise crucial desta etapa, entretanto, foi qualitativa, consistindo em ouvir o sinal inteiro sintetizado com excitação musical e, de acordo com o resultado, concluir se a aplicação do método LPC para VOCODER musical é viável.

### 3.2.2 Análise dos efeitos de quantização

Nesta etapa do experimento, foi repetida a análise realizada na seção 3.2.1.3 utilizando, porém, os sinais de entrada quantizados. Para o experimento, considerou-se as quantizações de 8 bits, 16 bits e 32 bits, buscando reproduzir arquiteturas de sistemas digitais comerciais.

Para cada tamanho de palavra, realizou-se a análise e síntese LPC do frame e comparou-se os resultados com a sintetização obtida sem aplicar quantização nos sinais.

De posse dos sinais sintetizados, definiu-se o erro de quantização realizando a subtração entre o sinal sintetizado com quantização e o sinal sintetizado sem quantização. Para garantir a coerência nas análises, normalizou-se a amplitude dos sinais sintetizados em relação à amplitude do sinal sintetizado sem quantização.

Por fim, tendo o sinal de erro, calculou-se a raiz quadrada do erro médio quadrático, ou valor RMS, do erro de quantização. Numa aplicação real, os sinais da fala e do instrumento são obtidos através de um conversor analógico-digital. Considerou-se então, para fins práticos, um conversor com excursão de -1 Volt a +1 Volt para o cálculo do erro RMS de quantização. A fórmula para cálculo do valor RMS do erro de quantização é descrita por

$$e_{rms}[n] = \sqrt{\frac{1}{N} \sum_{n=1}^N (x[n] - x_q[n])^2} \text{ Volts}, \quad (62)$$

onde  $x[n]$  é o sinal sintetizado sem quantização,  $x_q[n]$  é o sinal sintetizado com quantização e  $N$  é o tamanho do *frame* analisado.

Os resultados obtidos nesta etapa do trabalho são mostrados na seção 4.2.

## 4 RESULTADOS

### 4.1 VALIDAÇÃO DO MÉTODO DE LPC PARA APLICAÇÃO MUSICAL

#### 4.1.1 Análise LPC em *frame* de fala isolado.

Os resultados obtidos com o experimento descrito na seção 3.2.1.1 são mostrados a seguir.

Figura 26 – Frame de fala de 32ms.

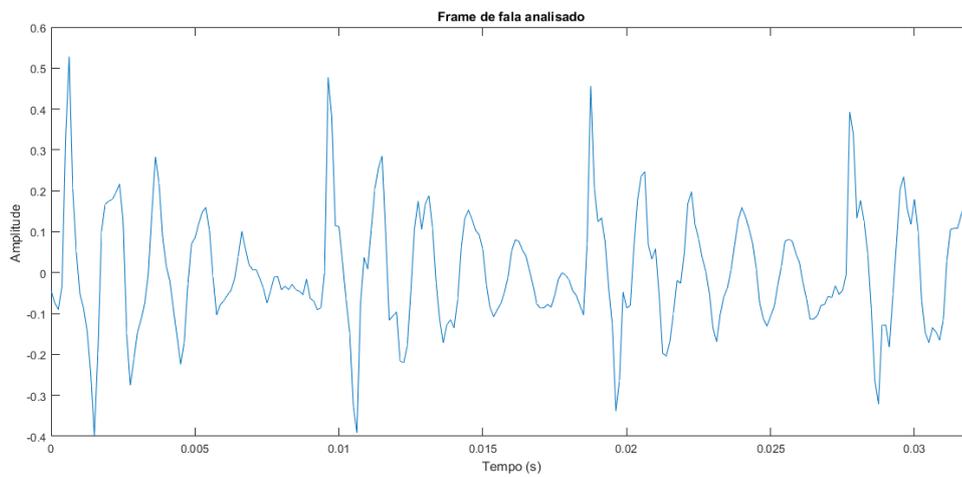


Figura 27 – Densidade espectral do *frame*.

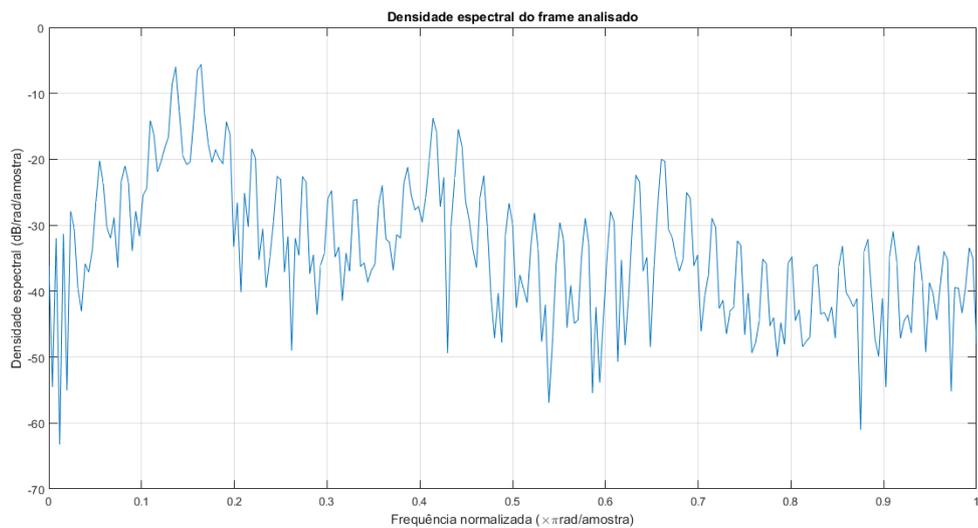


Tabela 1 – Coeficientes LPC do *frame* analisado.

$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$
-0,94788	0,514716	-0,19538	0,141126	0,135786
$\alpha_6$	$\alpha_7$	$\alpha_8$	$\alpha_9$	$\alpha_{10}$
-0,05073	0,489937	-0,12465	-0,27338	0,263998

Figura 28 – Resposta em frequência do filtro LPC.

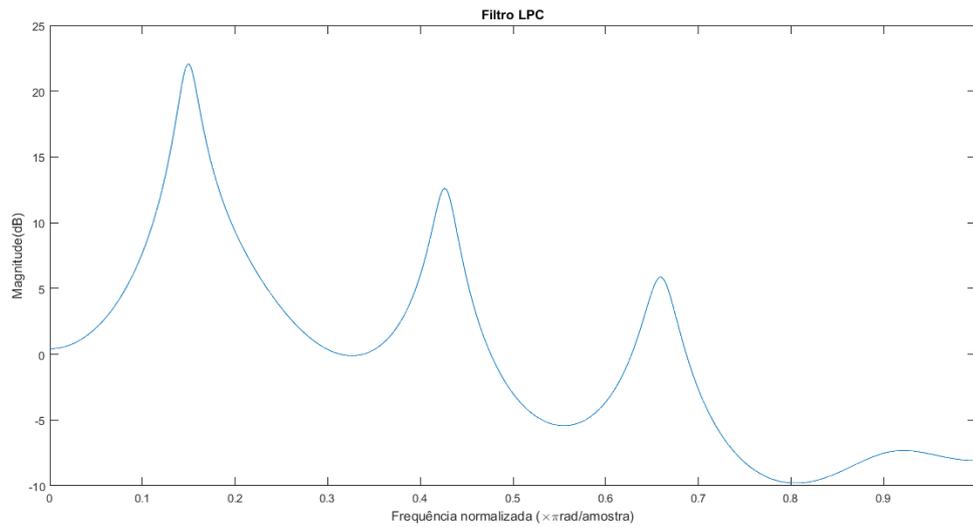


Figura 29 – Polos do filtro LPC no plano-z.

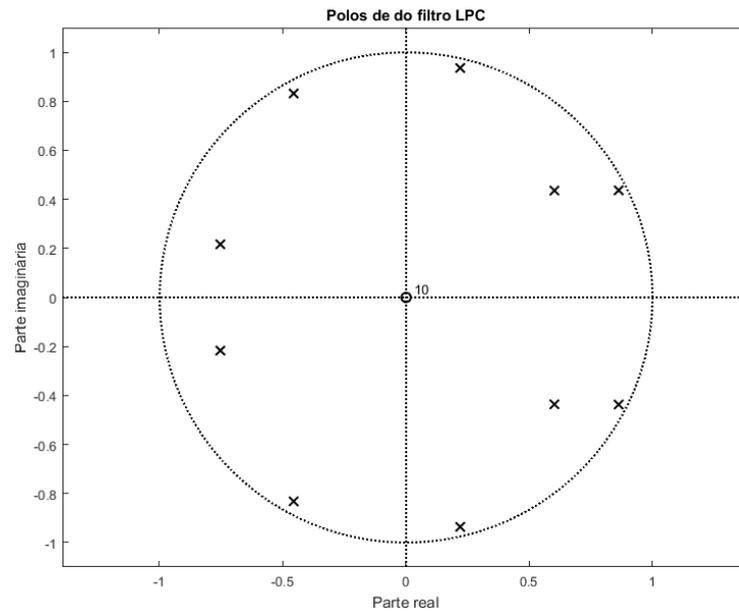
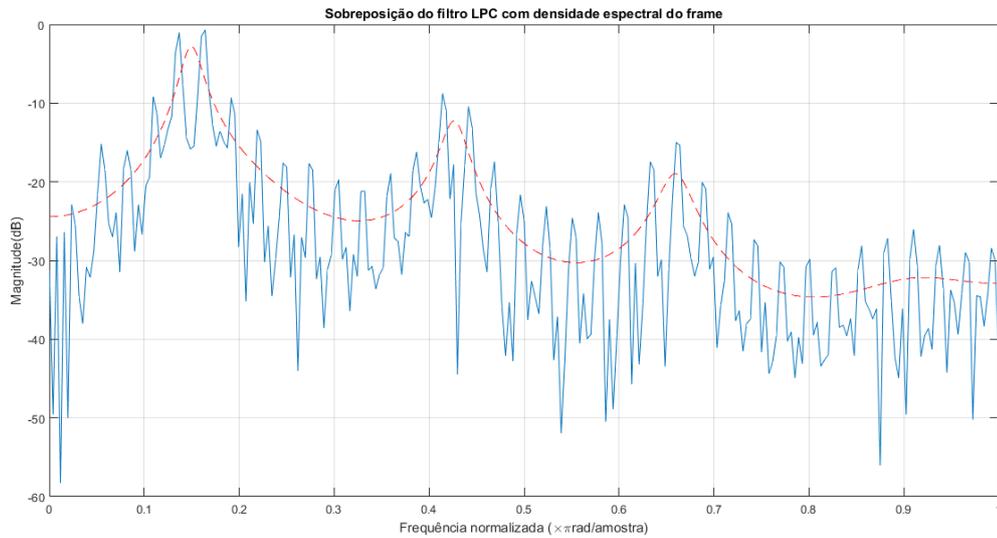


Figura 30 – Sobreposição do formato da resposta do filtro LPC à densidade espectral do *frame*.



O fato do formato da resposta em frequência do filtro LPC acompanhar o formato das formantes na densidade espectral do *frame* analisado é um indicativo de que o algoritmo de análise LPC funciona. Além disso, fica validada a utilização de 10 coeficientes de predição para o experimento.

#### 4.1.2 *Benchmark* do filtro LPC

Os resultados obtidos com o experimento descrito na seção 3.2.1.2 são mostrados a seguir.

Figura 31 – Trem de impulsos de 111 Hz aplicado como excitação para síntese LPC.

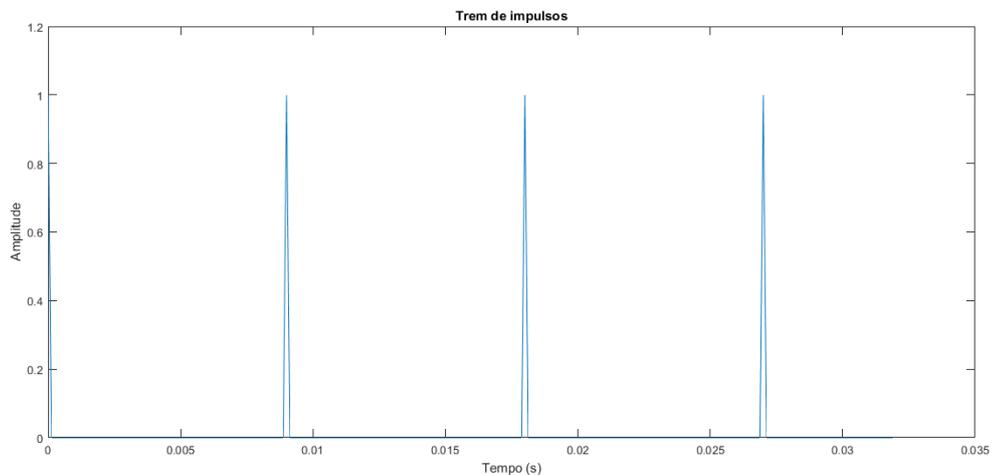


Figura 32 – Comparação entre *frame* original de fala e *frame* sintetizado.

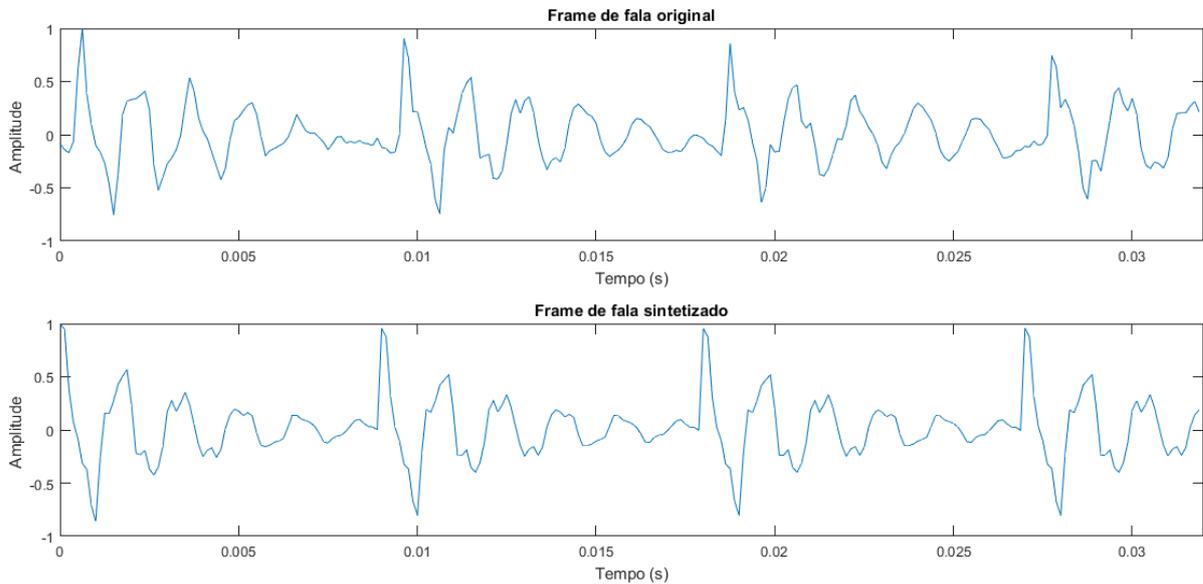
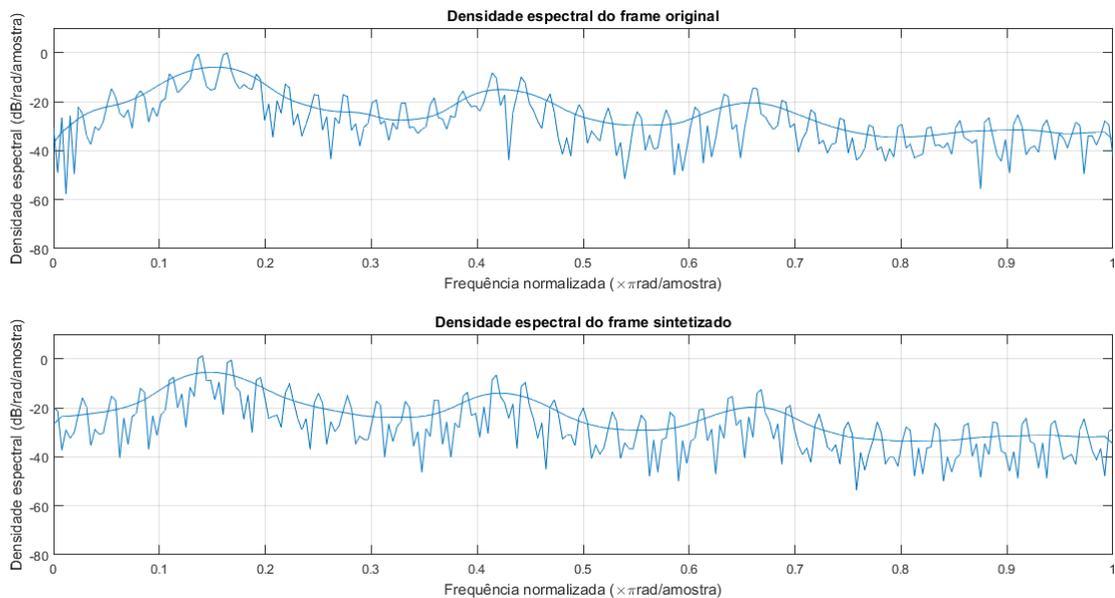


Figura 33 – Comparação entre densidade espectral do *frame* original e do *frame* sintetizado.



Tanto no domínio da frequência quanto no domínio do tempo, existe semelhança entre o sinal original e o sintetizado. Esta semelhança, entretanto, desaparece quando os gráficos são analisados em sua estrutura mais fina. No caso da densidade espectral do sinal, principalmente, percebe-se que o envelope espectral do sinal sintetizado e do sinal original são bastante semelhantes, entretanto, as componentes resultantes do sinal de excitação são diferentes. Esta diferença é coerente, uma vez que, no sinal original, a excitação ocorre por

um sinal de pulsos glotais, tal qual o mostrado na Figura 7 e, no sinal sintetizado, a excitação se dá por um trem de impulsos na frequência fundamental do *frame*.

Outro fator que deve ser observado é o fato de existir uma defasagem entre o sinal original e o sinal sintetizado, defasagem esta que impossibilita a subtração dos sinais para obtenção de um sinal de erro.

#### 4.1.3 Síntese LPC com sinal de excitação musical em *frame* isolado

Os resultados obtidos com o experimento descrito na seção 3.2.1.3 são mostrados a seguir.

Figura 34 – Densidade espectral do *frame* de excitação.

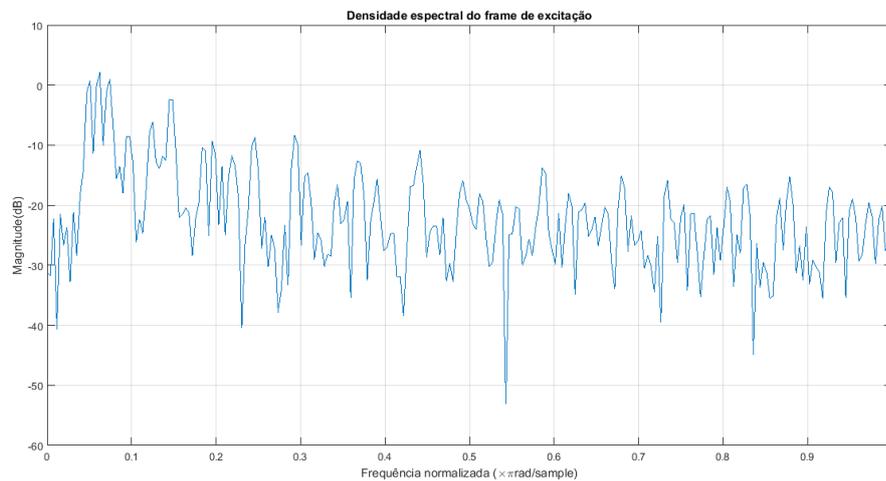
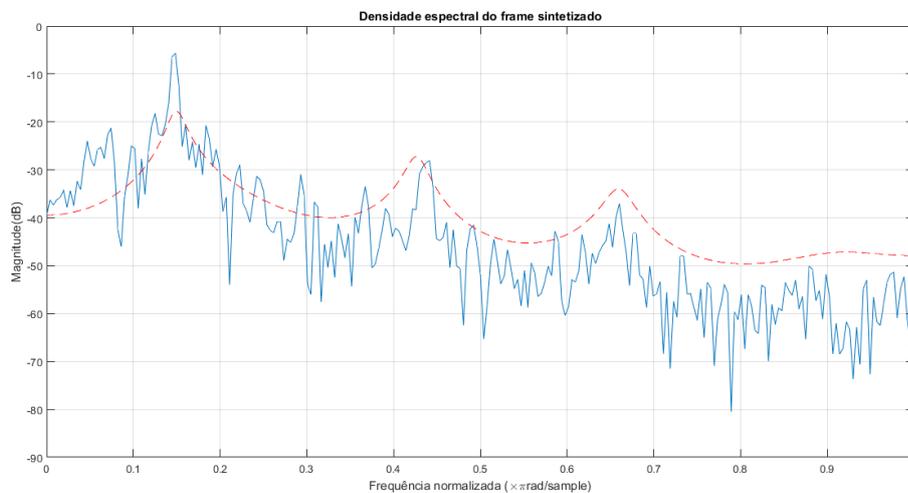


Figura 35 - Densidade espectral do *frame* sintetizado.



Nota-se, na Figura 35, que a densidade espectral do sinal sintetizado consiste da densidade espectral do sinal de excitação modulado pela resposta do filtro LPC. Este fato sugere sucesso no processo de síntese LPC para aplicação musical, uma vez que se tem a modulação do acorde com a resposta em frequência do trato vocal.

#### 4.1.4 Análise e síntese musical com sinal de fala inteiro.

Os resultados obtidos no experimento descrito na seção 3.2.1.4 são mostrados a seguir.

Figura 36 – Polos alocados no primeiro quadrante do plano-z.

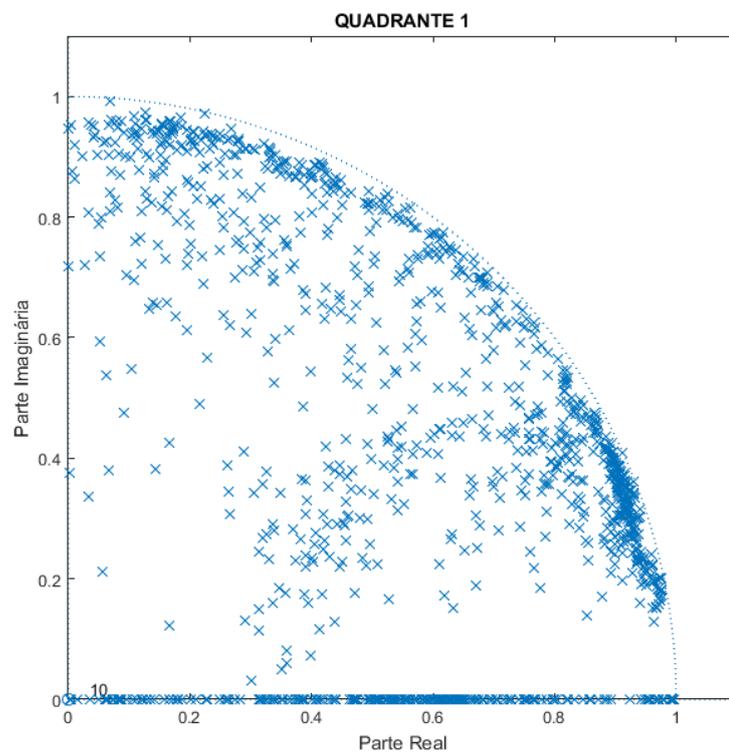


Figura 37 - Polos alocados no segundo quadrante do plano-z.

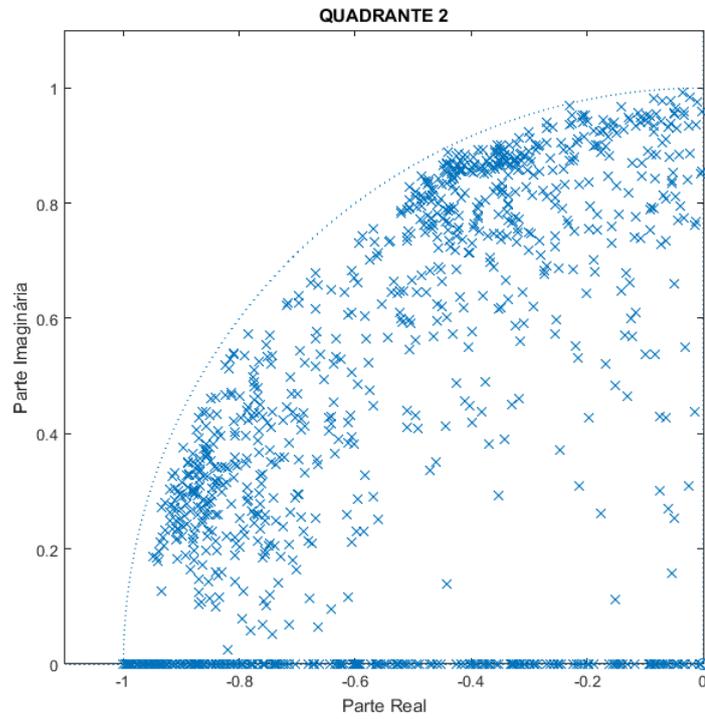


Figura 38 - Polos alocados no terceiro quadrante do plano-z.

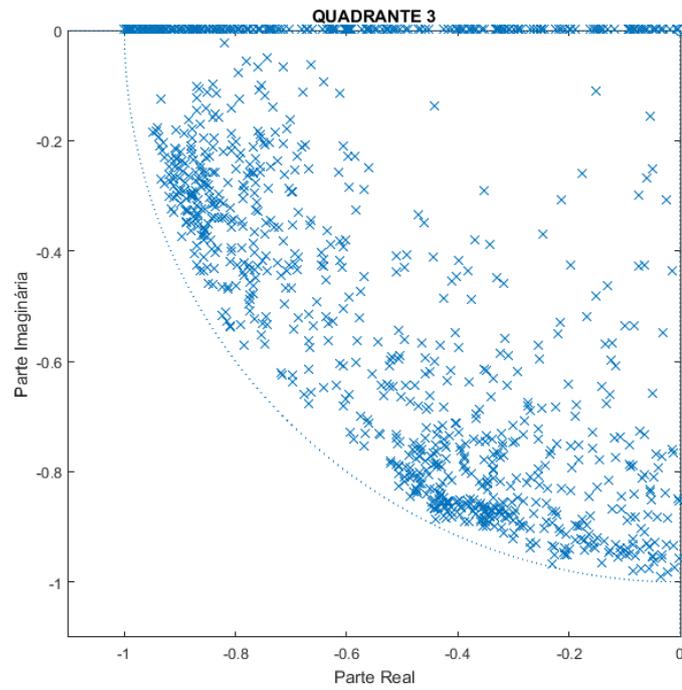
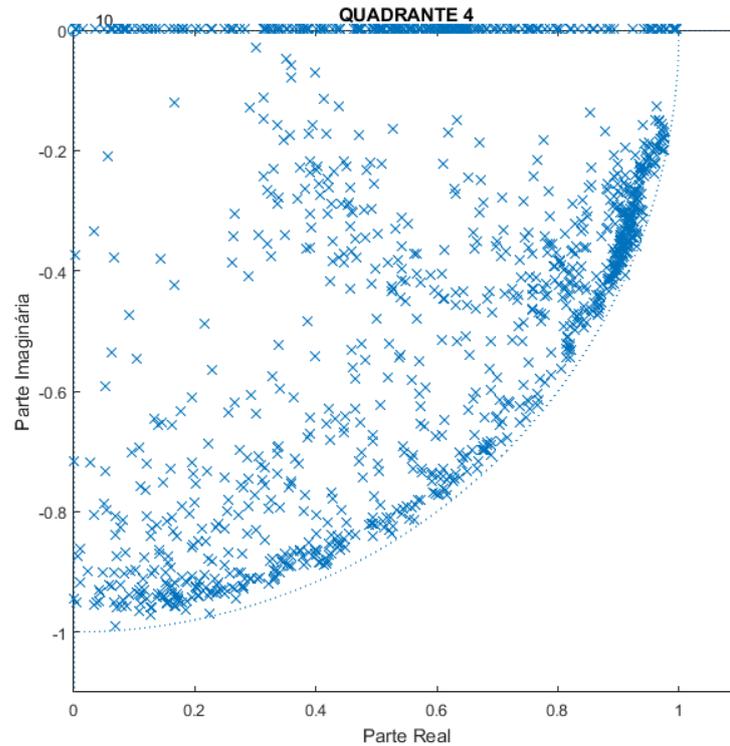


Figura 39 - Polos alocados no quarto quadrante do plano-z.



A inexistência de polos fora do círculo unitário sugere a estabilidade do método em todas as janelas de fala sintetizadas. Entretanto, o acúmulo de polos próximos ao círculo unitário sugere a possibilidade de instabilidade para o método quantizado, pois o arredondamento das suas posições pode posicioná-los fora da zona de estabilidade.

O resultado ouvido do sinal sintetizado foi satisfatório, dando à fala o aspecto esperado. Em suma, os sinais de entrada no sistema são uma gravação de fala e um acorde de Sol Maior e a saída é esta mesma fala “cantada” em Sol Maior.

## 4.2 ANÁLISE DOS EFEITOS DE QUANTIZAÇÃO NA ANÁLISE LPC

### 4.2.1 Quantização de 8 bits

A seguir, são mostrados os resultados do procedimento descrito na seção 3.2.2 utilizando quantização com um tamanho de palavra de 8 bits.

Figura 40 – Frames de fala e excitação quantizados com 8 bits e sua densidade espectral.

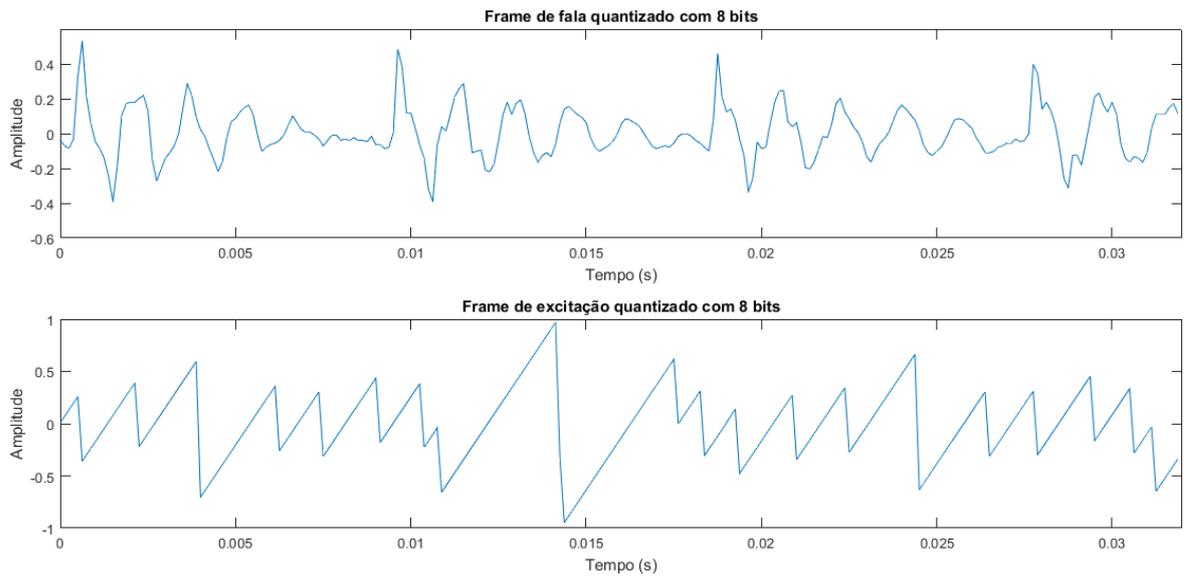


Tabela 2 - Coeficientes LPC do *frame* quantizado com 8 bits.

$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$
-0,70765	0,016554	0,067795	-0,11275	0,018087
$\alpha_6$	$\alpha_7$	$\alpha_8$	$\alpha_9$	$\alpha_{10}$
0,117541	0,105305	-0,06165	-0,18204	0,009093

Figura 41 – Filtro LPC frame quantizado com 8 bits.

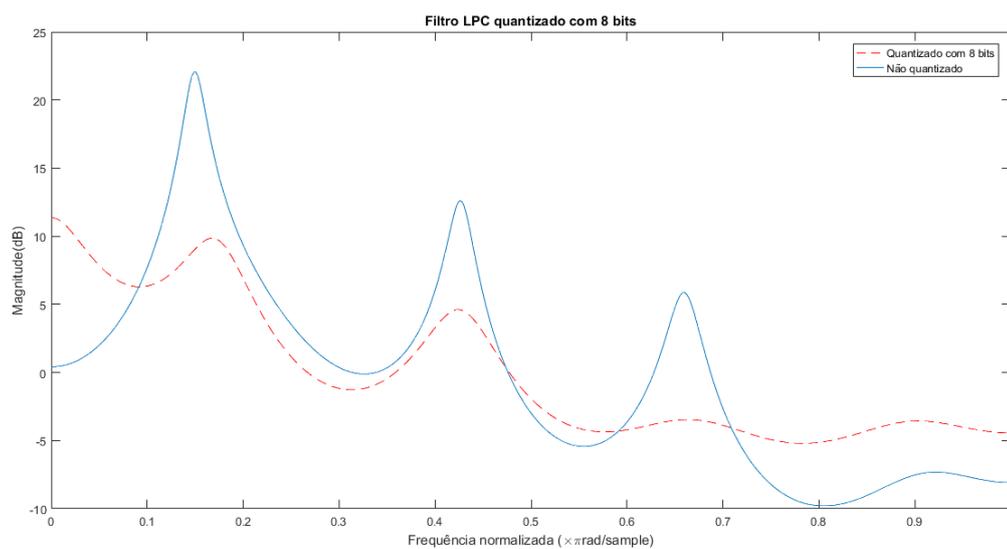
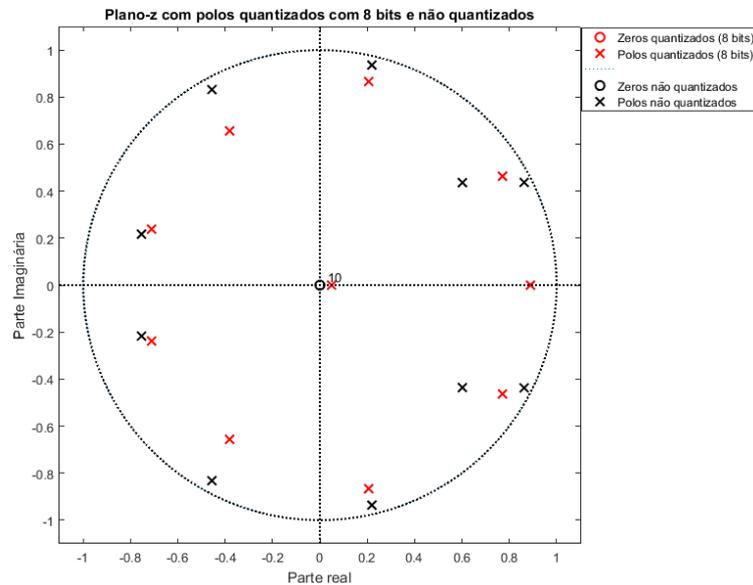


Figura 42 – Sobreposição de polos quantizados com 8 bits e não quantizados no plano-z



Neste ponto já fica claro que os erros referentes à quantização do sinal tornam a quantização de 8 bits inviável. Ao realizar a síntese com o sinal de excitação também quantizado com 8 bits, obtém-se uma saída cujo erro tem maior amplitude que o sinal sintetizado sem quantização, como mostra a Figura 43.

Figura 43 – Comparação entre *frame* sintetizado com 8 bits e sintetizado sem quantização.

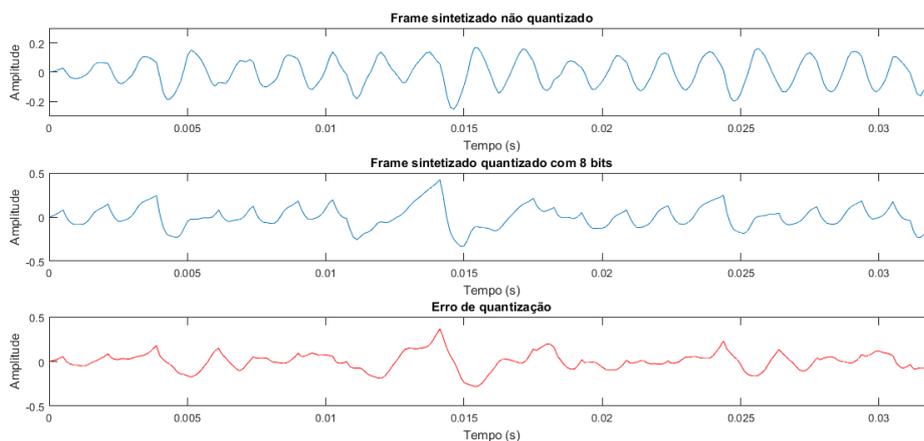
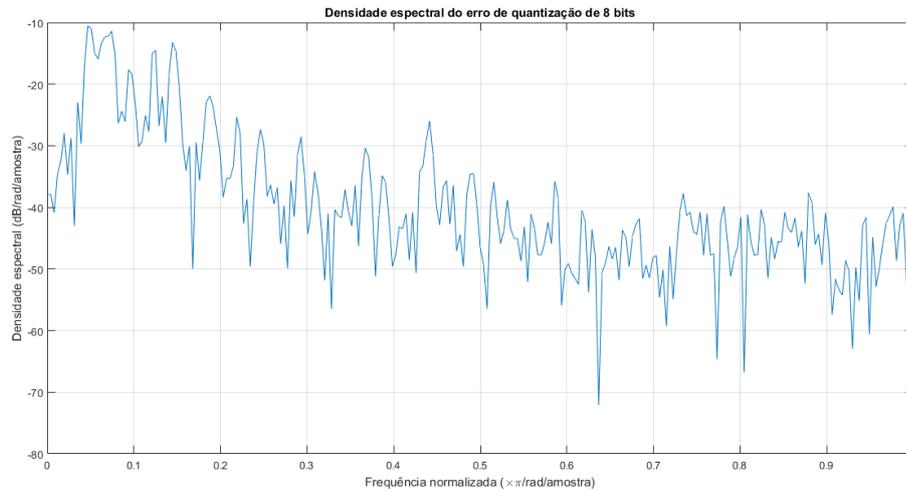


Figura 44 – Densidade espectral do erro de quantização com 8 bits.



Considerando que o sinal de fala e excitação são obtidos em um conversor A/D com faixa de -1 Volt a 1 Volt, o erro RMS de quantização obtido com 8 bits de tamanho de palavra é de 99 mV. Dividindo o erro de quantização RMS pelo valor RMS do sinal sintetizado não quantizado, obtém-se um erro RMS relativo de 105,65%, valor que confirma a inviabilidade da aplicação da quantização com tamanho de palavra de 8 bits.

#### 4.2.2 Quantização de 16 bits

A seguir, são mostrados os resultados do procedimento descrito na seção 3.2.2 utilizando quantização com um tamanho de palavra de 16 bits

Figura 45 – *Frames* de fala e excitação quantizados com 16 bits e sua densidade espectral.

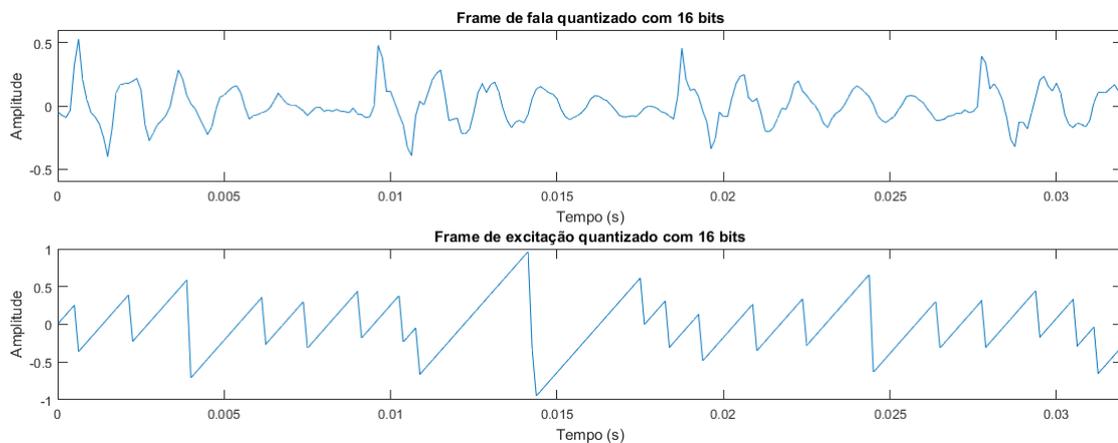


Tabela 3 - Coeficientes LPC do *frame* quantizado com 16 bits.

$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$
-0,98325	0,541967	-0,21964	0,146165	0,138275
$\alpha_6$	$\alpha_7$	$\alpha_8$	$\alpha_9$	$\alpha_{10}$
-0,0831	0,507964	-0,16152	-0,24986	0,236852

Figura 46 – Filtro LPC quantizado com 16 bits.

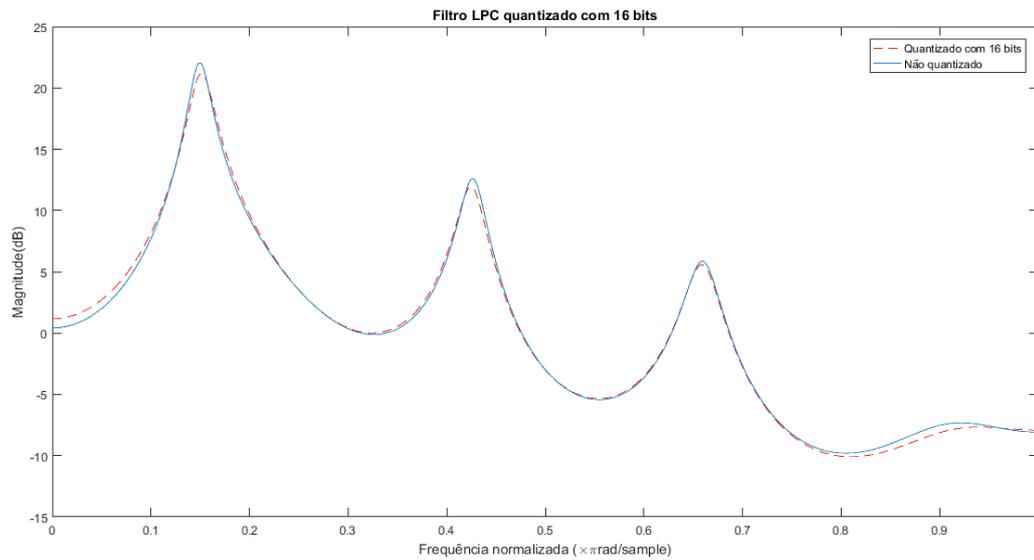


Figura 47 – Sobreposição de polos quantizados com 16 bits e não quantizados no plano-z.

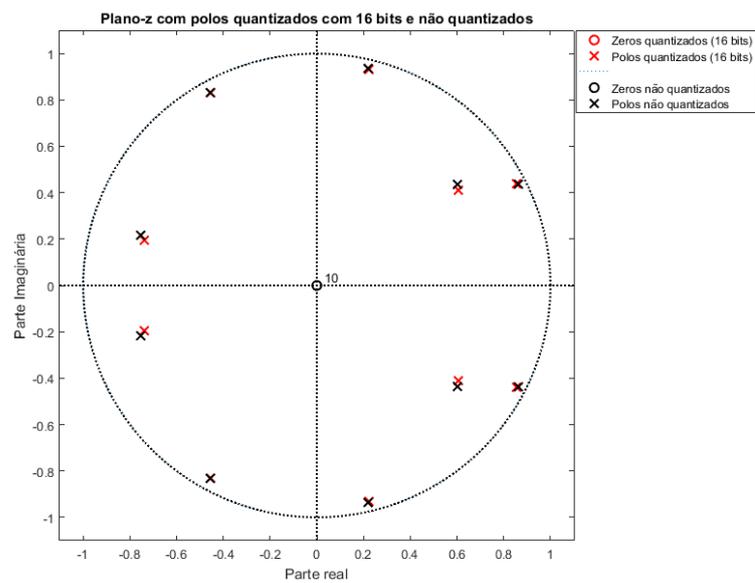


Figura 48 – Comparação entre frame sintetizado com 16 bits e sintetizado sem quantização.

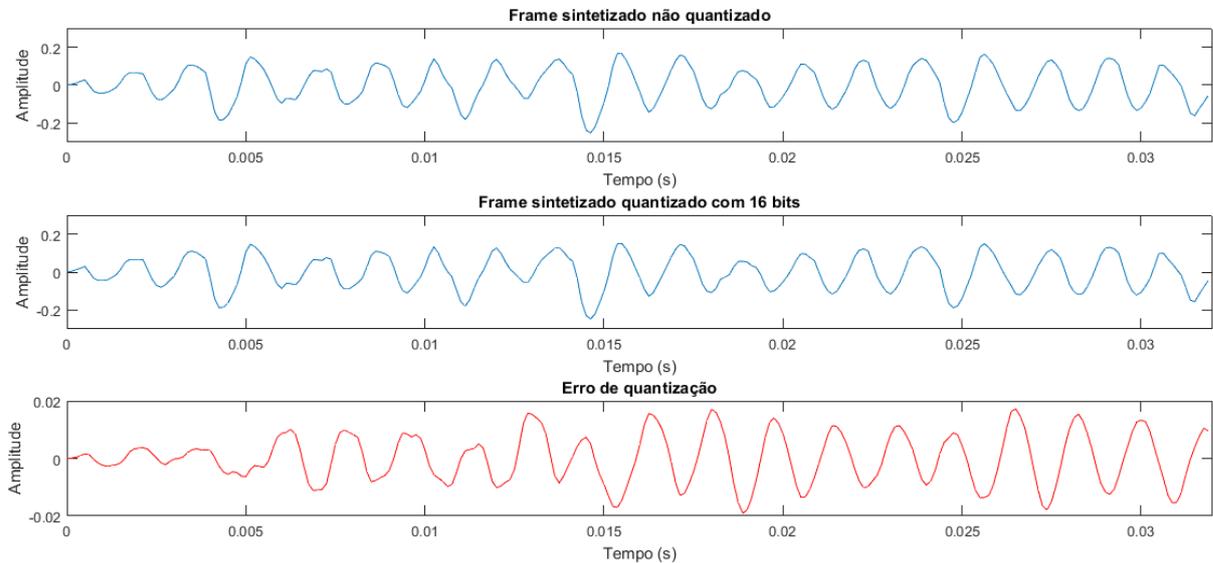
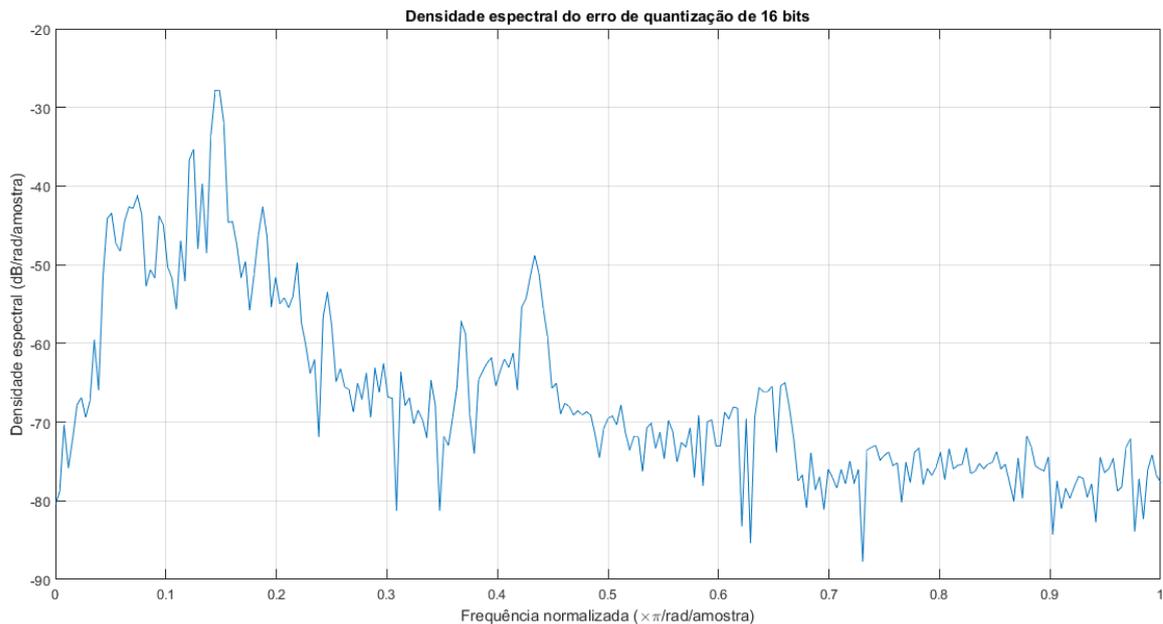


Figura 49 – Densidade espectral do erro de quantização de 16 bits.



Com uma precisão numérica de 16 bits, o sistema já apresenta um resultado muito próximo do caso sem quantização. Na Figura 47, todos os polos no plano  $z$  ficam muito próximos ao caso sem quantização, existindo, em algumas ocorrências, a sobreposição total de polos quantizados e não quantizados, impossibilitando sua distinção visual. A Figura 46

deixa claro o funcionamento do método, uma vez que a resposta do filtro quantizado praticamente se sobrepõe à resposta do filtro não quantizado. A densidade de espectro máxima do erro de quantização, conforme mostra a Figura 49, fica na casa de -38dB.

Considerando que o sinal de fala e excitação são obtidos em um conversor A/D com faixa de -1 Volt a 1 Volt, o erro RMS de quantização obtido com a 16 bits de tamanho de palavra é de 8,7mV. Dividindo o erro de quantização RMS pelo valor RMS do sinal sintetizado não quantizado, obtém-se um erro RMS relativo de 9,17%.

#### 4.2.3 Quantização de 32 bits

A seguir, são mostrados os resultados do procedimento descrito na seção 3.2.2 utilizando quantização com um tamanho de palavra de 32 bits.

Figura 50 – Frame de fala quantizado com 32 bits e sua densidade espectral.

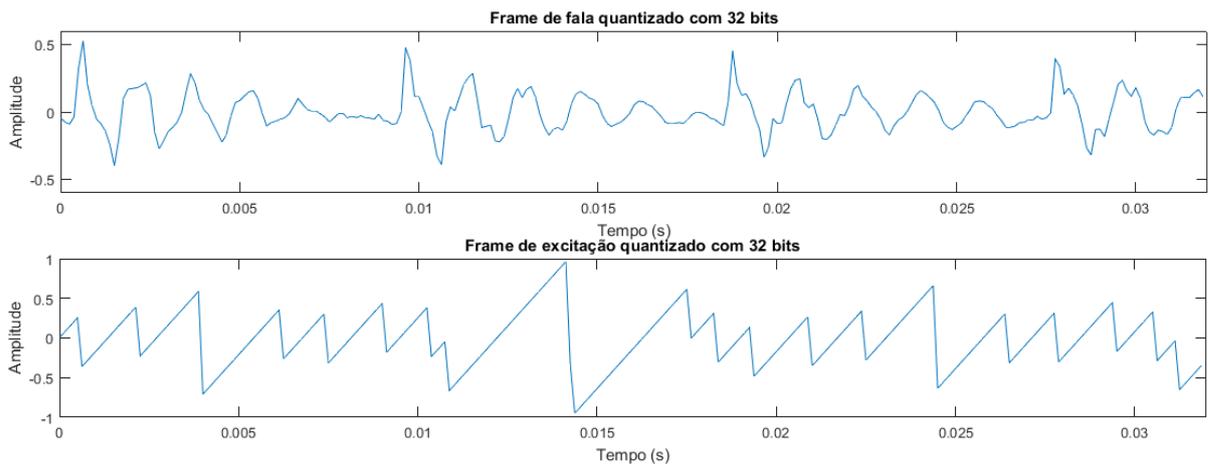


Tabela 4 - Coeficientes LPC do *frame* quantizado com 32 bits.

$\alpha_1$	$\alpha_2$	$\alpha_3$	$\alpha_4$	$\alpha_5$
-0,9851	0,542592	-0,21968	0,145269	0,13916
$\alpha_6$	$\alpha_7$	$\alpha_8$	$\alpha_9$	$\alpha_{10}$
0,234749	0,336335	-0,26208	-0,14272	0,146739

Figura 51 – Filtro LPC quantizado com 32bits.

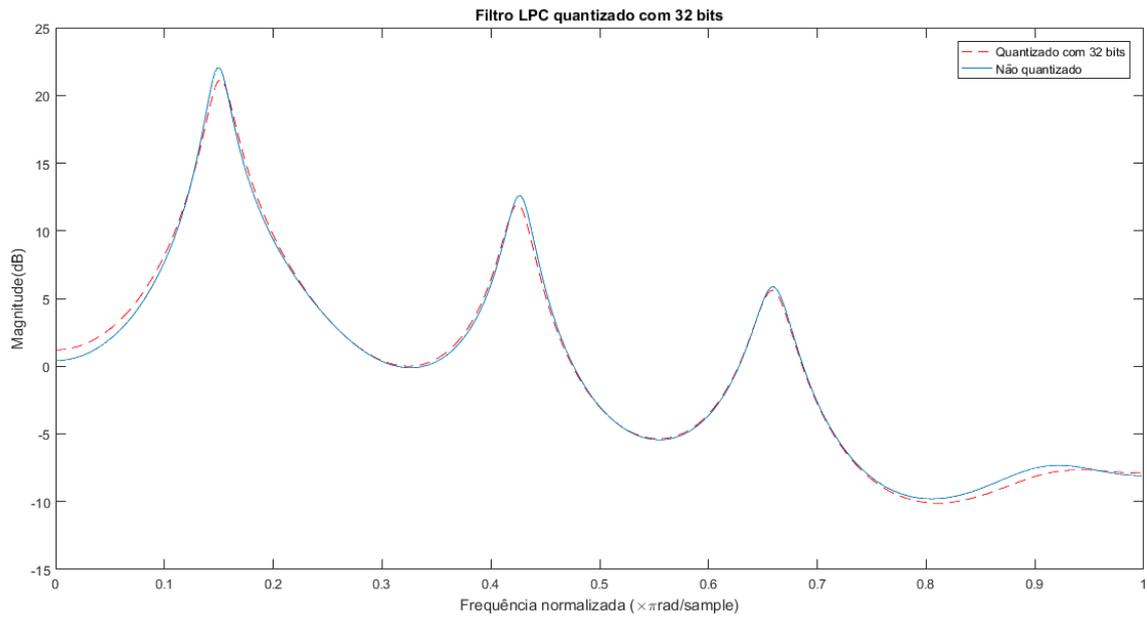


Figura 52 – Sobreposição de polos quantizados com 32 bits e não quantizados no plano-z.

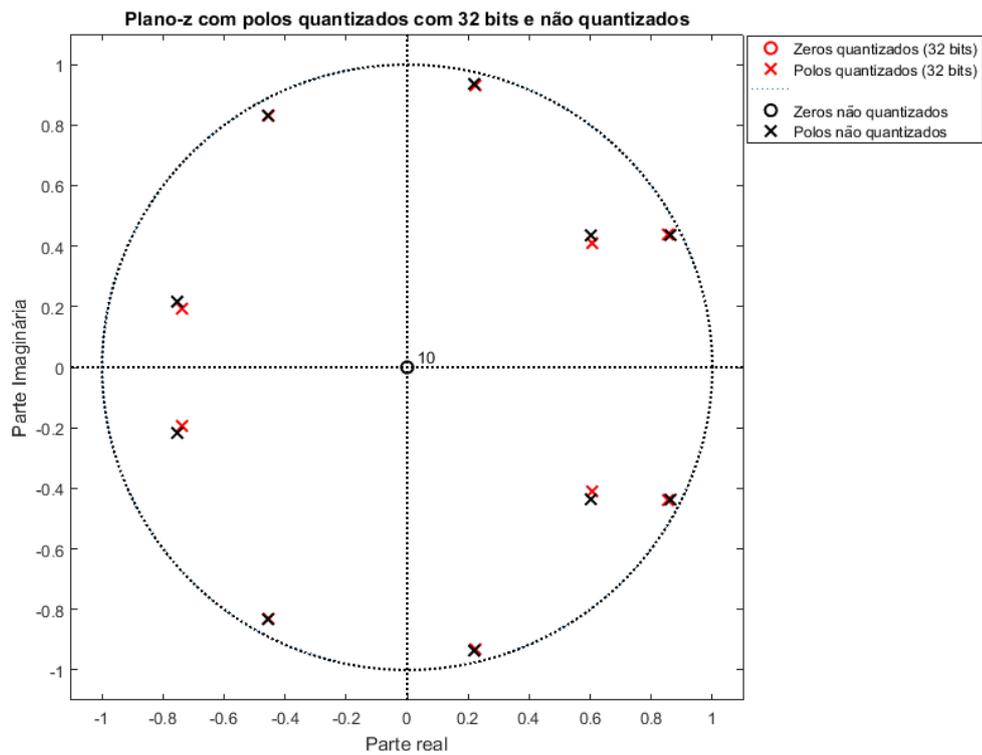


Figura 53 – Comparação entre frame sintetizado com 32 bits e sintetizado sem quantização.

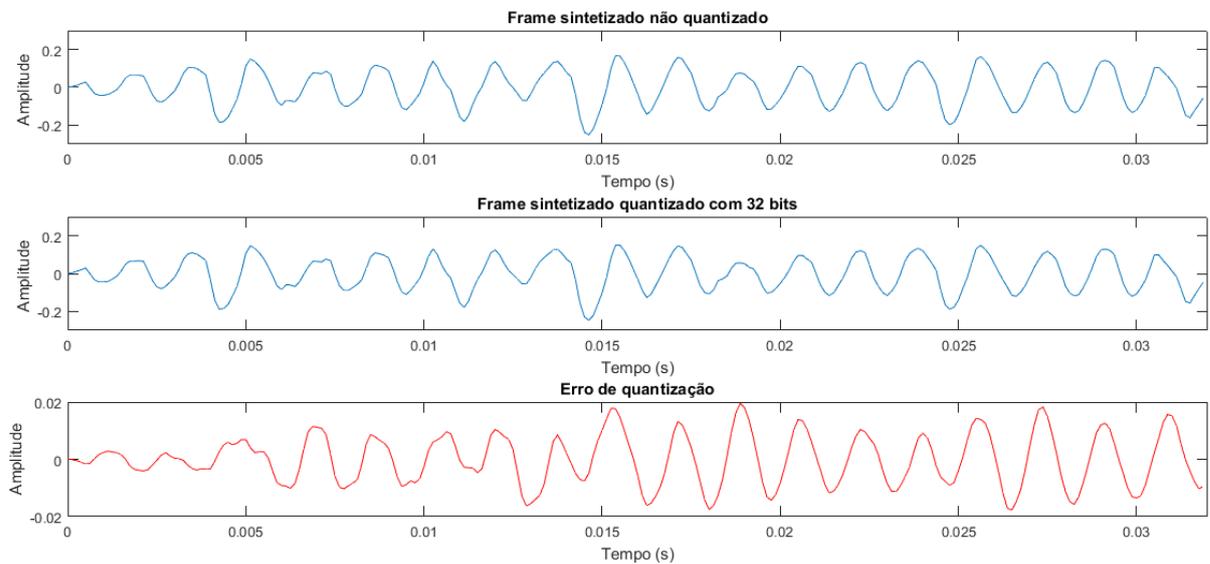
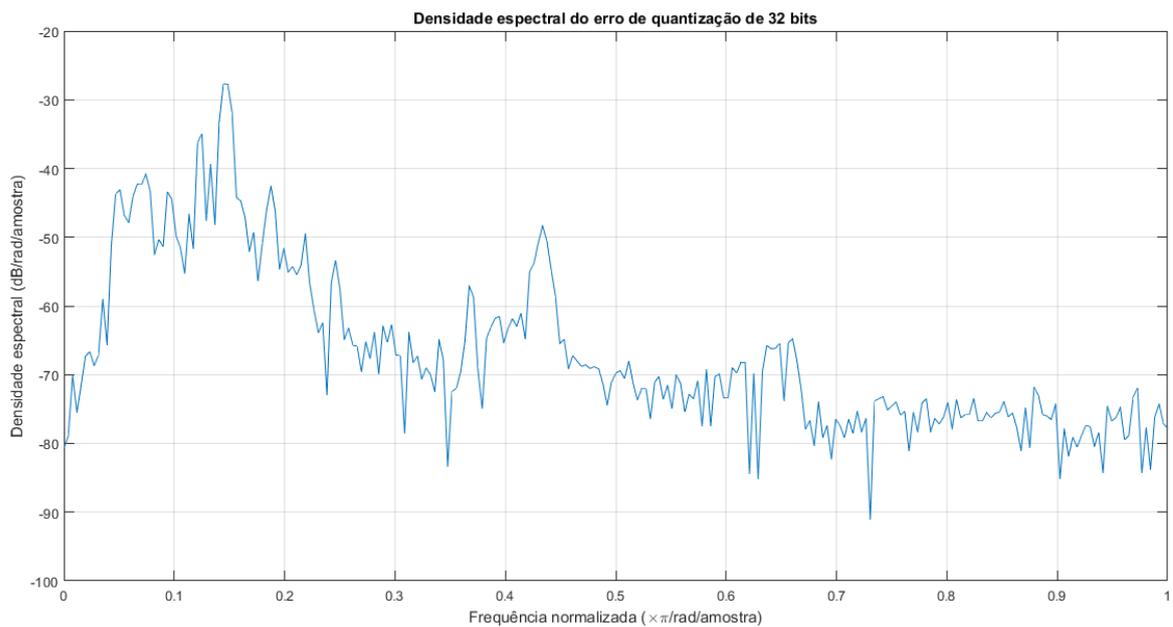


Figura 54 – Densidade espectral do erro da quantização de 32 bits.



Com 32 bits, o resultado é muito semelhante ao obtido com a quantização de 16 bits, não justificando, inclusive, a utilização desta resolução. A máxima densidade espectral do ruído fica da casa dos -28dB. A resposta em frequência do filtro LPC fica idêntica ao caso da quantização com 16 bits.

Considerando que o sinal de fala e excitação são adquiridos em um conversor A/D com faixa de -1 Volt a 1 Volt, o erro RMS de quantização obtido com a 32 bits de tamanho de

palavra é de 8,1 mV. Dividindo o erro de quantização RMS pelo valor RMS do sinal sintetizado não quantizado, obtém-se um erro RMS relativo de 8,56%.

## 5 CONCLUSÃO

Os resultados dos experimentos buscando a prova de conceito do método LPC para aplicação musical se mostraram promissores. Na análise de *frame* único, utilizando 10 coeficientes de predição, os filtros gerados pelo método seguiram o formato das formantes do *frame* da fala. Este fato viabiliza a premissa de modular o instrumento musical com as características do trato vocal.

Ao realizar o *benchmark* do filtro, buscando a reconstrução do sinal de entrada, fica clara a discrepância do sinal sintetizado em relação ao original em sua estrutura mais fina. Existe, entretanto, muita semelhança entre os envelopes das densidades espectrais dos sinais.

Aplicando um sinal musical como excitação do sistema, a resposta em frequência do *frame* sintetizado mantém as características das frequências formantes da fala, modulando o sinal do instrumento musical com a resposta em frequência do trato vocal. Ao realizar a análise LPC de um sinal de fala inteiro e aplicar um acorde musical no filtro proveniente desta análise, o resultado confirmou a viabilidade do método aplicação musical, dando “dicção” ao acorde.

Com isso, ouvindo o resultado da síntese musical utilizando LPC, fica confirmada a viabilidade do método para o VOCODER musical. Cabe, entretanto, a experimentação com outros tipos de sinal de excitação e instrumentos para verificar a versatilidade do método e buscar diferentes sinais que apresentam resultados agradáveis do ponto de vista musical.

Em relação aos efeitos da quantização com ponto fixo, o único procedimento que não apresentou resultados satisfatórios foi a quantização utilizando 8 bits, onde a variação na posição dos polos tornou o sinal sintetizado irreconhecível em relação ao caso não quantizado. As quantizações com 16 bits e 32 bits apresentaram erros de quantização ínfimos, indicando viabilidade no procedimento. Assim, para implementação do vocoder em plataformas DSP de ponto fixo, do ponto de vista de precisão numérica, um tamanho de palavra de 16 bits já é suficiente.

Entretanto, testes adicionais devem ser feitos para definição da plataforma DSP a ser utilizada. Uma análise crucial que deve ser feita em trabalhos futuros é o tempo de processamento por iteração, uma vez que o objetivo do VOCODER, quando implementado, é que o sinal sintetizado tenha o menor atraso possível em relação às entradas da fala e do instrumento, permitindo sua utilização em música ao vivo. Aqui, vale testar outros tipos de

recursão para obtenção dos parâmetros de predição da forma mais rápida possível e também a variação da quantidade de coeficientes de predição utilizados, buscando o valor otimizado do ponto de vista de resultado e de custo computacional

Ainda, para trabalhos futuros, recomenda-se a aplicação de ponto fixo no algoritmo de recursão para obtenção dos coeficientes LPC e do algoritmo de sintetização do sinal de excitação, para garantir uma avaliação exata da viabilidade do método inteiramente quantizado. A alternativa de execução do código com partes em ponto flutuante, apesar de simplificar a implementação e permitir uma maior precisão numérica, compromete o tempo de processamento em plataformas que não contam com arquitetura dedicada à esse tipo de operação.

Em suma, pode-se afirmar que a utilização do método LPC é aplicável para VOCODER musical, sendo interessante a exploração de outras fontes de excitação além da onda triangular. Ainda, pode-se afirmar que, em relação à precisão numérica, a implementação em plataformas de ponto fixo é viável, existindo, entretanto, a necessidade de análise do desempenho.

## 6 REFERÊNCIAS BIBLIOGRÁFICAS

ALAN V. OPPENHEIM, R. W. S. J. R. B. **Discrete-time Signal Processing**. 2ª. ed. [S.l.]: Prentice Hall, 1999.

CHILDERS, D. G. **Speech Processing and Synthesis Toolboxes**. [S.l.]: Wiley, 2000.

FANT, G. **Acoustic Theory of Speech Production: With Calculations based on X-Ray Studies of Russian Articulations**. [S.l.]: Walter de Gruyter, v. II, 1971.

FURUI, S. **Digital Speech Processing: Synthesis and Recognition**. 2ª. ed. [S.l.]: Taylor & Francis, 2000.

JOHN G. PROAKIS, D. G. M. **Digital Signal Processing**. 4ª. ed. [S.l.]: Pearson Prentice Hall, 2007.

KAPLAN, W. **Advanced Calculus**. 5ª. ed. [S.l.]: Addison-Wesley, 2002.

MAKHOUL, J. Linear prediction: A tutorial review. **Proceedings of the IEEE**, Abril 1975. 561 - 580.

PETER LADEFOGED, K. J. **A Course in Phonetics**. 6ª. ed. [S.l.]: Cengage Learning, 2010.

ROTHENBERG, M. The Glottal Volume Velocity Waveform During Loose and. **Proceedings of the Seventh International Congress of Phonetic Sciences**, 22-28 Agosto 1971.

SIMON HAYKIN, B. V. V. **Signals and Systems**. 2ª. ed. [S.l.]: Wiley India Pvt. Limited, 2007.

THIERRY DUTOIT, F. M. **Applied Signal Processing: A MATLAB™-Based Proof of Concep**. [S.l.]: Springer Science & Business Media, 2010.

TOMPKINS, D. **How to Wreck a Nice Beach: The Vocoder from World War II to Hip-Hop, The Machine Speaks**. [S.l.]: Melville House, 2011.

VISSER, F. Le Vocodeur d'Elektor. **Elektor**, Fevereiro 1980.

VOCODER VP-03. **Site da Roland**, 2017. Disponível em:  
<<https://www.roland.com/us/products/vp-03/>>. Acesso em: 13 Julho 2017.

## ANEXO A – ALGORITMO PARA EXPERIMENTO COM FRAME ÚNICO

```

carrega arquivo de fala -> amostragem de 8k
[fala,fs] = audioread('male.wav');
n = 0:length(fala);
%definição do frame de 256 amostras (32ms)
L = 256;
n=0:L-1;
t=n./fs;
inicio = 9765;
frame = fala(inicio:inicio+L-1);
figure;
plot(t,frame);
xlabel('Tempo (s)'); ylabel('Amplitude normalizada');
title('Frame de fala analisado');
xlim([0 0.032]);
figure;
periodogram(frame,[],512);
xlabel('Frequência normalizada (\times\pi rad/amostra)');
ylabel('Densidade espectral (dB/rad/amostra)');
title('Densidade espectral do frame analisado');
%analise lpc do frame
[coef, sigma_square] = lpc(frame, 10);
sigma = sqrt(sigma_square);
%filtro LPC
[H,W]=freqz(1,coef, 512);
figure;
plot(W/pi,20*log10(abs(H)),'-');
title('Filtro LPC');
xlabel('Frequência normalizada (\times\pi rad/amostra)');
ylabel('Magnitudo(dB)');
%polos no plano z
figure;
zplane(1,coef);
title('Polos de do filtro LPC');
xlabel('Parte real');
ylabel('Parte imaginária');
%sobreposição do filtro de síntese com o periodograma do frame;
figure;
periodogram(frame,[],512,2);
hold on;
plot(W/pi,20*log10(sigma*abs(H)),'r--');
title('Sobreposição do filtro LPC com densidade espectral do frame');
xlabel('Frequência normalizada (\times\pi rad/amostra)');
ylabel('Magnitudo(dB)');
hold off;

%carrega arquivo de fala
%define vetor 'n'
%define tamanho do frame
%redefine tamanho do vetor n
%redefine tamanho do vetor t
%define posição inicial do frame

%plota frame no domínio do tempo

%plota densidade espectral do frame

%realiza análise LPC com 10 coeficientes
%desvio padrão do resíduo
%obtem a resposta em frequência
%do filtro LPC
%utilizando os coeficientes obtidos
%na análise LPC do frame

%plota os polos do filtro LPC
%no plano z

%BENCHMARK DO FILTRO LPC
N=72
for i=1:1000
    excit((i*N)-(N-1):i*N) = [1;zeros((N-1),1)];
end

%define frame
%de excitação
%de 256 amostras

%aplica frame de
%excitação no filtro LPC

%normaliza a amplitude
%dos sinais para posterior
%comparação

%plota gráficos
figure;
plot(t,excit_frame);
title('Trem de impulsos');
xlabel('Tempo (s)'); ylabel('Amplitude normalizada');
ylim([0 1.2]);
figure;
subplot(2,1,1);

```

```

plot(t,frame);
title ('Frame de fala original');
xlabel('Tempo (s)'); ylabel('Amplitude normalizada');
xlim([0 0.032]);
subplot(2,1,2);
plot(t,synt_frame);
title ('Frame de fala sintetizado');
xlabel('Tempo (s)'); ylabel('Amplitude normalizada');
xlim([0 0.032]);
figure;
subplot(2,1,1);
periodogram(frame,[],512);
hold on;
pwelch(frame);
xlabel('Frequência normalizada (\times\pi\text{rad}/amostra)');
ylabel('Densidade espectral (dB/rad/amostra)');
title ('Densidade espectral do frame original');
ylim([-80 10]);
hold off;
subplot(2,1,2);
periodogram(synt_frame,[],512);
hold on;
pwelch(synt_frame);
xlabel('Frequência normalizada (\times\pi\text{rad}/amostra)');
ylabel('Densidade espectral (dB/rad/amostra)');
title ('Densidade espectral do frame sintetizado');
ylim([-80 10]);
hold off;

```

#### %EXCITAÇÃO COM ACORDE MUSICAL

```

% sinal musical
n=0:length(fala)-1;
t=n./fs;
nota = sawtooth(2*pi*196*t) + sawtooth(2*pi*247*t) + sawtooth(2*pi*294*t); %cria o acorde de excitação
nota=nota./abs(max(nota)); %normaliza amplitude do sinal de excitação
n=0:L-1; %redefine tamanho do vetor n
t=n./fs; %redefine tamanho do vetor t
%frame de excitação
frame_exc = nota(inicio:inicio+L-1); %define frame de excitação
figure; %com o mesmo tamanho do frame
plot(t,frame_exc); %de fala analisado
xlabel('Tempo (s)'); ylabel('Amplitude normalizada');
xlim([0 0.032]);
figure;
periodogram(frame_exc,[],512);
title ('Densidade espectral do frame de excitação');
xlabel('Frequência normalizada (\times\pi\text{rad}/sample)');
ylabel('Magnitudo(dB)');
%passagem do sinal de excitação pelo filtro LPC
synt_frame=filter(sigma,coef,frame_exc); %filtra o frame de excitação
figure; %com o filtro LPC obtido do frame
plot(t,synt_frame); %de fala e plota o resultado nos
xlabel('Tempo (s)'); ylabel('Amplitude normalizada'); %domínios do tempo e da frequência
xlim([0 0.032]);
figure;
periodogram(synt_frame,[],512);
hold on;
plot(w/pi,20*log10(sigma*abs(H))-15,'r--');
title ('Densidade espectral do frame sintetizado');
xlabel('Frequência normalizada (\times\pi\text{rad}/sample)');
ylabel('Magnitudo(dB)');

```

## ANEXO B – ALGORITMO PARA EXPERIMENTO COM SINAL INTEIRO

```

[fala,fs] = audioread('male.wav'); %carrega sinal de fala
fala = fala(1:length(fala)/8); %encurta sinal de fala para 6,37s
fala_synt = zeros(length(fala),1); %cria buffer
n = 0:length(fala)-1; %define vetor n do tamanho do sinal de fala
t = n/fs; %cria vetor t
nota = sawtooth(2*pi*196*t) + sawtooth(2*pi*247*t) + sawtooth(2*pi*294*t); %cria sinal de excitação
nota = transpose(nota); %transpõe nota para a mesma orientação da fala
L=256; %define tamanho da janela
overlap = L/2; %define overlap como 50%
N = (length(fala)-(L-overlap))/overlap; %define quantidade de janelas total
p=10; %define quantidade de coeficientes LPC

%%INICIO DO LOOP DE ANALISE E SÍNTESE
for i=1:N-1 %repete loop até a última janela
    inicio = ((1+(i-1)*(L - overlap))); %posição de início da janela em função da
    %iteração
    fim = inicio + L-1; %posição final da janela

    input_frame = fala(inicio:fim).*hann(L); %janelamento do sinal de fala na posição correta
    excitation = nota(inicio:fim).*hann(L); %janelamento do sinal de excitação na posição
    %correta

    [ai, sigma_square]=lpc(input_frame,p); %faz análise LPC
    sigma=sqrt(sigma_square); %define desvio padrão do resíduo->ganho

    synt_frame=filter(sigma, ai,excitation); %faz síntese do sinal com o filtro LPC obtido
    slice = fala_synt(inicio:fim) + synt_frame; %realiza overlap
    fala_synt(inicio:fim) = slice;

    coeficientes((i*(p+1))-p:i*(p+1)) = ai; %acumula coeficientes LPC
end
sound (fala_synt);

%PLOTAGENS
figure;
for k=1:N
    coef = coeficientes((k*(p+1)-p:k*(p+1)));
    [H,W] = freqz(1,coef,512); %resposta de todos filtros
    plot(w/pi,20*log10(abs(H)))
    hold on;
end
xlabel('Frequência normalizada (\times\pi rad/sample)');
ylabel('Magnitude(dB)');

figure;
for k=1:N
    zplane(1,coeficientes((k*11)-10:11*k)); %plota todos os polos
    hold on; %no quadrante 1
end
xlim([0 1.1]);
ylim([0 1.1]);
xlabel('Parte Real');
ylabel('Parte Imaginária');
title('QUADRANTE 1')

figure;
for k=1:N
    zplane(1,coeficientes((k*11)-10:11*k)); %plota todos os polos
    hold on; %no quadrante 2
end
xlim([-1.1 0]);
ylim([0 1.1]);
xlabel('Parte Real');
ylabel('Parte Imaginária');
title('QUADRANTE 2')

figure;

```

```
for k=1:N
zplane(1,coeficientes((k*11)-10:11*k));           %plota todos os polos
hold on;                                         %no quadrante 3
end
xlim([-1.1 0]);
ylim([-1.1 0]);
xlabel('Parte Real');
ylabel('Parte Imaginária');
title('QUADRANTE 3')

figure;
for k=1:N
zplane(1,coeficientes((k*11)-10:11*k));           %plota todos os polos
hold on;                                         %no quadrante 4
end
xlim([0 1.1]);
ylim([-1.1 0]);
xlabel('Parte Real');
ylabel('Parte Imaginária');
title('QUADRANTE 4')
```

## ANEXO C – ALGORITMO PARA EXPERIMENTO COM QUANTIZAÇÃO

```

%carrega arquivo de fala
[fala,fs] = audioread('male.wav');
n=0:length(fala)-1;
t=n./fs;
nota = sawtooth(2*pi*196*t) + sawtooth(2*pi*247*t) + sawtooth(2*pi*294*t);
nota=nota./abs(max(nota));
n=0:255;
t=n./fs;
inicio = 9765;
frame = fala(inicio:inicio+255);
excit = nota(inicio:inicio+255);
[ai, sigma_square] = lpc(frame, 10);
sigma = sqrt(sigma_square);
[HI,WI]=freqz(ai, 1, 512);
[H,W]=freqz(1,ai, 512);
synt_frame=filter(sigma,ai,excit);

synt_quad = (synt_frame).^2;
synt_med_quad = sum(synt_quad)/256;
synt_RMS = sqrt(synt_med_quad)

%%converte os sinais de entrada para ponto fixo
fala = fi(fala);
nota = fi(nota);

%quantização com 8 bits
w1 = 8;
f1 = w1-1;
rm = 'Ceiling';
fala_quant_8 = quantize(fala,1,w1,f1,rm);
nota_quant_8 = quantize(nota,1,w1,f1,rm);
n=0:255;
t=n./fs;
inicio = 9765;
frame_8 = fala_quant_8(inicio:inicio+255);
excit_8 = nota_quant_8(inicio:inicio+255);
[ai_8, sigma_square_8] = lpc_fixo(frame_8, 10);
sigma_8 = sqrt(sigma_square_8);
[HI_8,WI_8]=freqz(ai_8, 1, 512);
[H_8,W_8]=freqz(1,ai_8, 512);
synt_frame_8=filter(double(sigma_8),double(ai_8),double(excit_8));
% synt_frame_8_norm=synt_frame_8./max(abs(synt_frame_8));
% synt_frame_norm=synt_frame;

erro_8 = synt_frame_8 - synt_frame;

erro_quad_8 = (erro_8).^2;
erro_med_quad_8 = sum(erro_quad_8)/256;
erro_RMS_8 = sqrt(erro_med_quad_8);
erro_relativo_8 = erro_RMS_8/synt_RMS;

%repete procedimento do trecho anterior mas quantizando com 16 bits
w1 = 16;
f1 = w1-1;
rm = 'Ceiling';
fala_quant_16 = quantize(fi(fala),1,w1,f1,rm);
nota_quant_16 = quantize(fi(nota),1,w1,f1,rm);
n=0:255;
t=n./fs;
inicio = 9765;
frame_16 = fala_quant_16(inicio:inicio+255);
excit_16 = nota_quant_16(inicio:inicio+255);
[ai_16, sigma_square_16] = lpc_fixo(frame_16, 10);
sigma_16 = sqrt(sigma_square_16);
[HI_16,WI_16]=freqz(ai_16, 1, 512);
[H_16,W_16]=freqz(1,ai_16, 512);
synt_frame_16=filter(double(sigma_16),double(ai_16),double(excit_16));
% synt_frame_16_norm=synt_frame_16./max(abs(synt_frame_16));

```

```

% synt_frame_norm=synt_frame./max(abs(synt_frame)); %para posterior comparação
erro_16 = synt_frame_16 - synt_frame;
erro_quad_16 = (erro_16).^2;
erro_med_quad_16 = sum(erro_quad_16)/256;
erro_RMS_16 = sqrt(erro_med_quad_16);
erro_relativo_16 = erro_RMS_16/synt_RMS;

%repete procedimento do trecho anterior mas quantizando com 32 bits
w1 = 32;
f1=w1-1;
rm = 'Ceiling';
fala_quant_32 = quantize(fi(fala),1,w1,f1,rm);
nota_quant_32 = quantize(fi(nota),1,w1,f1,rm);
n=0:255;
t=n./fs;
inicio = 9765;
frame_32 = fala_quant_32(inicio:inicio+255);
excit_32 = nota_quant_32(inicio:inicio+255);
[ai_32, sigma_square_32] = lpc_fixo(frame_32, 10);
sigma_32 = sqrt(sigma_square_32);
[HI_32,WI_32]=freqz(ai_32, 1, 512);
[H_32,w_32]=freqz(1,ai_32, 512);
synt_frame_32=filter(double(sigma_32),double(ai_32),double(excit_32));
synt_frame_32_norm=synt_frame_32./max(abs(synt_frame_32)); %normaliza sinais
synt_frame_norm=synt_frame./max(abs(synt_frame)); %para comparação do
erro_32 = synt_frame - synt_frame_32; %das sintetizações com
%e sem quantização

erro_quad_32 = (erro_32).^2;
erro_med_quad_32 = sum(erro_quad_32)/256;
erro_RMS_32 = sqrt(erro_med_quad_32);
erro_relativo_32 = erro_RMS_32/synt_RMS;
%%plota resultados da quantização de 8 bits
figure;
subplot(2,1,1);
plot(t,frame_8);
title('Frame de fala quantizado com 8 bits');
xlabel('Tempo (s)'); ylabel('Amplitude');
xlim([0 0.032])
subplot(2,1,2);
plot(t,excit_8);
xlim([0 0.032])
title('Frame de excitação quantizado com 8 bits');
xlabel('Tempo (s)'); ylabel('Amplitude');
figure;
subplot(311),
plot(t,synt_frame);
title('Frame sintetizado não quantizado');
xlabel('Tempo (s)'); ylabel('Amplitude');
xlim([0 0.032]);
subplot(312);
plot(t,synt_frame_8);
title('Frame sintetizado quantizado com 8 bits');
xlabel('Tempo (s)'); ylabel('Amplitude');
xlim([0 0.032]);
subplot(313);
plot(t,erro_8,'r');
title('Erro de quantização');
xlabel('Tempo (s)'); ylabel('Amplitude');
xlim([0 0.032]);
figure;
periodogram(erro_8,[],512);
title('Densidade espectral do erro de quantização de 8 bits');
xlabel('Frequência normalizada (\times\pi/rad/amostra)'); ylabel('Densidade espectral (dB/rad/amostra)')
figure;
[z_8 p_8 e] = zplane(1,ai_8);
hold on;
[z p e] = zplane(1,ai);
set(findobj(p_8, 'Type', 'line'), 'color', 'r','linewidth',1.5);
set(findobj(z_8, 'Type', 'line'), 'color', 'r','linewidth',1.5);
set(findobj(p, 'Type', 'line'), 'color', 'black','linewidth',1.5);
set(findobj(z, 'Type', 'line'), 'color', 'black','linewidth',1.5);
set(findobj(e, 'Type', 'line'), 'color', 'black','linewidth',1.5)
xlim([-1.1 1.1]);
ylim([-1.1 1.1]);
xlabel('Parte real'),ylabel('Parte Imaginária');

```

```

legend('Zeros quantizados (8 bits)', 'Polos quantizados (8 bits)', '', 'Zeros não quantizados', 'Polos não quantizados');
title('Plano-z com polos quantizados com 8 bits e não quantizados')
hold off;
figure;
plot(w_8/pi, 20*log10(abs(H_8)), 'r--', w/pi, 20*log10(abs(H)), '-');
title('Filtro LPC quantizado com 8 bits');
xlabel('Frequência normalizada (\times\pi rad/sample)');
ylabel('Magnitude(dB)');
legend('Quantizado com 8 bits', 'Não quantizado');

```

```

%%plota resultados da quantização de 16 bits

```

```

figure;
subplot(2,1,1);
plot(t, frame_16);
title('Frame de fala quantizado com 16 bits');
xlabel('Tempo (s)'); ylabel('Amplitude');
xlim([0 0.032])
subplot(2,1,2);
plot(t, excit_16);
xlim([0 0.032])
title('Frame de excitação quantizado com 16 bits');
xlabel('Tempo (s)'); ylabel('Amplitude');
figure;
subplot(311),
plot(t, synt_frame);
title('Frame sintetizado não quantizado');
xlabel('Tempo (s)'); ylabel('Amplitude');
xlim([0 0.032]);
subplot(312);
plot(t, synt_frame_16);
title('Frame sintetizado quantizado com 16 bits');
xlabel('Tempo (s)'); ylabel('Amplitude');
xlim([0 0.032]);
subplot(313);
plot(t, erro_16, 'r');
title('Erro de quantização');
xlabel('Tempo (s)'); ylabel('Amplitude');
xlim([0 0.032]);
figure;
periodogram(erro_16, [], 512);
title('Densidade espectral do erro de quantização de 16 bits');
xlabel('Frequência normalizada (\times\pi rad/amostra)'); ylabel('Densidade espectral (dB/rad/amostra)');
figure;
[z_16 p_16 e] = zplane(1, ai_16);
hold on;
[z p e] = zplane(1, ai);
set(findobj(p_16, 'Type', 'line'), 'Color', 'r', 'linewidth', 1.5);
set(findobj(z_16, 'Type', 'line'), 'Color', 'r', 'linewidth', 1.5);
set(findobj(p, 'Type', 'line'), 'Color', 'black', 'linewidth', 1.5);
set(findobj(z, 'Type', 'line'), 'Color', 'black', 'linewidth', 1.5);
set(findobj(e, 'Type', 'line'), 'Color', 'black', 'linewidth', 1.5);
xlim([-1.1 1.1]);
ylim([-1.1 1.1]);
xlabel('Parte real'), ylabel('Parte Imaginária');
legend('Zeros quantizados (16 bits)', 'Polos quantizados (16 bits)', '', 'Zeros não quantizados', 'Polos não quantizados'); title('Plano-z com polos quantizados com 16 bits e não quantizados')
hold off;
figure;
plot(w_16/pi, 20*log10(abs(H_16)), 'r--', w/pi, 20*log10(abs(H)));
title('Filtro LPC quantizado com 16 bits');
xlabel('Frequência normalizada (\times\pi rad/sample)');
ylabel('Magnitude(dB)');
legend('Quantizado com 16 bits', 'Não quantizado');

```

```

%%plota resultados da quantização de 32 bits

```

```

figure;
subplot(2,1,1);
plot(t, frame_32);
title('Frame de fala quantizado com 32 bits');
xlabel('Tempo (s)'); ylabel('Amplitude');
xlim([0 0.032])
subplot(2,1,2);
plot(t, excit_32);
title('Frame de excitação quantizado com 32 bits');

```

```

xlabel('Tempo (s)'); ylabel('Amplitude');
xlim([0 0.032])
figure;
subplot(311),
plot(t,synt_frame);
title('Frame sintetizado não quantizado');
xlabel('Tempo (s)'); ylabel('Amplitude ');
xlim([0 0.032]);
subplot(312);
plot(t,synt_frame_32);
title('Frame sintetizado quantizado com 32 bits');
xlabel('Tempo (s)'); ylabel('Amplitude ');
xlim([0 0.032]);
subplot(313);
plot(t,erro_32,'r');
title('Erro de quantização');
xlabel('Tempo (s)'); ylabel('Amplitude');
xlim([0 0.032]);
figure;
periodogram(erro_32,[],512);
title('Densidade espectral do erro de quantização de 32 bits');
xlabel('Frequência normalizada (\times\pi/rad/amostra)'); ylabel('Densidade espectral (dB/rad/amostra)');
figure;
[z_32 p_32 e] = zplane(1,ai_32);
hold on;
[z p e] = zplane(1,ai);
set(findobj(p_32, 'Type', 'line'), 'color', 'r','linewidth',1.5);
set(findobj(z_32, 'Type', 'line'), 'color', 'r','linewidth',1.5);
set(findobj(p, 'Type', 'line'), 'color', 'black','linewidth',1.5);
set(findobj(z, 'Type', 'line'), 'color', 'black','linewidth',1.5);
set(findobj(e, 'Type', 'line'), 'color', 'black','linewidth',1.5);
xlim([-1.1 1.1]);
ylim([-1.1 1.1]);
xlabel('Parte real'),ylabel('Parte Imaginária');
legend('Zeros quantizados (32 bits)','Polos quantizados (32 bits)',','Zeros não quantizados','Polos não quantizados');title('Plano-z com polos quantizados com 16 bits e não quantizados')
title('Plano-z com polos quantizados com 32 bits e não quantizados')
hold off;
figure;
plot(w_32/pi,20*log10(abs(H_32)), 'r--',w/pi,20*log10(abs(H)));
title ('Filtro LPC quantizado com 32 bits');
xlabel('Frequência normalizada (\times\pi/rad/sample)');
ylabel('Magnitude(dB)');
legend('Quantizado com 32 bits','Não quantizado');

```